iimasNLP at SemEval-2024 Task 8: Unveiling structure-aware language models for automatic generated text identification

Andric Valdez,
Posgrado en Ciencia e Ingeniería de la Computación
Helena Gomez-Adorno, Fernando Márquez, Jorge Pantaleón,
Instituto de Investigaciones en Matemáticas y en Sistemas
Gemma Bel-Enguix
Instituto de Ingeniería
Universidad Nacional Autónoma de México, Ciudad de México
andric_valdez@comunidad.unam.mx

Abstract

Large language models (LLMs) are artificial intelligence systems that can generate text, translate languages, and answer questions in a human-like way. While these advances are impressive, there is concern that LLMs could also be used to generate fake or misleading content. In this work, as a part of our participation in SemEval-2024 Task-8, we investigate the ability of LLMs to identify whether a given text was written by a human or by a specific AI. We believe that human and machine writing style patterns are different from each other, so integrating features at different language levels can help in this classification task. For this reason, we evaluate several LLMs that aim to extract valuable multilevel information (such as lexical, semantic, and syntactic) from the text in their training processing. Our best scores on SubtaskA (monolingual) and SubtaskB were 71.5% and 38.2% in accuracy, respectively (both using the ConvBERT LLM); for both subtasks, the baseline (RoBERTa) achieved an accuracy of 74%.

1 Introduction

Large language models (LLMs) have become widely available and easily accessible, leading to an increase in machine-generated content across diverse platforms including question-and-answer forums, social media platforms, educational resources, and academic settings.

Recent advancements in LLM technology, exemplified by models like ChatGPT and GPT-4, produce coherent responses to a vast majority of user inquiries, making them increasingly appealing for replacing human labor in various applications. However, this accessibility has raised concerns about potential misuse, such as generating fake news, financial services industry, legal domain, and disruptions in educational settings. Given the challenge humans face in distinguishing between machine-generated and human-written text, there is a pressing need to develop automated systems capable of identifying machine-generated content to mitigate the risks associated with its misuse.

Motivated by these challenges, SemEval-2024 Task-8 (Wang et al., 2024) offers three subtasks over two paradigms of text generation: (1) full text when a considered text is entirely written by a human or generated by a machine; and (2) mixed text when a machine-generated text is refined by a human or a human-written text paraphrased by a machine.

These three subtasks are composed in the following way: Subtask A is a binary classification task that focuses on identity if a given text was written by a human or a machine; it is split into monolingual (English) and multilingual (Arabic, Russian, Chinese, etc). Subtask B is a multi-class classification task that aims to identify which specific LLM generates a given text among six different known options: Human-made, ChatGPT, Cohere, DaVinci, Bloomz, and Dolly. Finally, Subtask C, given a mixed text, where the first part is human-written and the second part is machine-generated, determines the boundary, where the change occurs.

We tackled two of these three subtasks: Subtask A (monolingual) and Subtask B. We applied fine-tuning of four LLMs (described in the following section) that included structural information in their pre-training. These models have proven their efficiency in multiple Natural Language Understanding (NLU) tasks, such as question-answer entailment, paraphrasing, and textual similarity. We aim to test the efficiency in machine-text detection by comparing the results of given baselines for each subtask (A and B) with our fine-tuning LLMs with different approaches for the implementation of structural information.

Our scores show a modest performance related to the final ranking (especially in Subtask B), but, based on the analysis of the results We observe that all of these LLMs used in this research, struggle

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1110–1114

to classify human text, meanwhile, they achieve a good performance classifying machine text.

This paper is structured as follows: Section 2 summarizes related works on machine text generation. Section 3 describes the dataset used for the task. Section 4 presents the system overview and the experimental setup. Section 5 and 6 shows the results and conclusions, respectively.

2 Related Work

In recent years, many interesting shared tasks that related to the automatic detection of AI-generated text. Besides the SemEval task8, one of the most popular and challenging tasks called Autextification: Automated Text Identification (Sarvazyan et al., 2023), aims to address the detection of content created by text generation models in English and Spanish.

To mention a few interesting research works related to Autextification-2023, the system titled "I've Seen Things You Machines Wouldn't Believe: Measuring Content Predictability to Identify Automatically Generated Text" (Przybyła et al., 2023) achieves the best performance among the submissions in subtask 1 (differentiating between human- and machine-generated text), both for English and Spanish. Their model focuses on assessing the "predictability" of given text by multiple LLMs, leveraging features related to grammatical accuracy, word frequency, and linguistic patterns, along with a fine-tuned LLM representation. Another remarkable work titled "Generative AI Text Classification using Ensemble LLM Approaches" (Abburi et al., 2023), proposes an ensemble neural model that leverages probabilities generated by different pre-trained LLMs as features for a Traditional Machine Learning (TML) classifier (their model ranked in first place in subtask 2 for English and Spanish).

On the other hand, pre-training LLMs with structural information enrich the learning process with contextual and syntactic cues. These cues encompass sentence structure, paragraph organization, grammatical rules, and broader linguistic patterns. Fine-tuning LLMs with such structural knowledge enhances their ability to both comprehend and generate text that adheres to human-like writing styles and conventions.

This approach has been explored in multiple ways; so now we briefly describe the approach taken by the models we used: ERNIE model (Sun et al., 2021) implements an implicit knowledge of syntactic information through multiple levels of masking (token, phrase, and entity level). SpanBERT model (Joshi et al., 2020) masks random spans of contiguous tokens and trains to predict every token for each span instead of just masking and predicting each token. ConvBERT model (Jiang et al., 2020) substitutes attention blocks for span-based dynamic convolutions capable of storing structural information in the generated kernels. Finally, XLNet (Yang et al., 2019), this LLM does not corrupt the text with masking but rather utilizes all the multiple permutations of tokens in a given sentence during the training process.

3 Dataset

The data provided for SemEval Task 8 is an extension of the M4 dataset (Wang et al., 2023). This is a large-scale benchmark, which is a multi-generator, multi-domain, and multi-lingual corpus for machine-generated text detection. This extensive M4 corpus encompasses texts from various domains, including news articles, programming code, and fictional narratives. Additionally, the M4 corpus incorporates texts in numerous languages, such as English, Spanish, and Chinese. This diversity in both domain and language coverage contributes to the effectiveness of M4 in effectively identifying machine-generated text (see figure 1).

For machine generation, it prompts the following multilingual LLMs: GPT-4, ChatGPT, GPT3.5 (tex-davinci-003), Cohere, and Dolly-v2. The models are asked to write articles given a title (Wikipedia), abstracts given a paper title (arXiv), peer reviews based on the title and the abstract of a paper (PeerRead), news briefs based on a title (news), also to summarize Wikipedia articles (Arabic), and to answer questions (Reddit).

Source/ Domain	Language	Total Human	Parallel Data Human	Davinci003	ChatGPT	Cohere	Dolly-v2	BLOOMz	Total
Wikipedia	English	6,458,670	3,000	3,000	2,995	2,336	2,702	3,000	17,033
Reddit ELI5	English	558,669	3,000	3,000	3,000	3,000	3,000	3,000	18,000
WikiHow	English	31,102	3,000	3,000	3,000	3,000	3,000	3,000	18,000
PeerRead	English	5,798	5,798	2,344	2,344	2,344	2,344	2,344	17,518
arXiv abstract	English	2,219,423	3,000	3,000	3,000	3,000	3,000	3,000	18,000
Baike/Web QA	Chinese	113,313	3,000	3,000	3,000	–	–	–	9,000
RuATD	Russian	75,291	3,000	3,000	3,000	–	–	–	9,000
Urdu-news	Urdu	107,881	3,000	–	3,000	–	–	–	9,000
id_newspapers_2018	Indonesian	499,164	3,000	–	3,000	–	–	–	6,000
Arabic-Wikipedia	Arabic	1,209,042	3,000	–	3,000	–	–	–	6,000
True & Fake News	Bulgarian	94,000	3,000	3,000	3,000	–	–	–	9,000
Total			35,798	23,344	32,339	13,680	14,046	14,344	133,551

Figure 1: Statistics about our M4 dataset, which includes non-parallel human data and parallel human and machine-generated texts.

4 System overview

Our system evaluates different LLMs that integrate features at different language levels (such as lexical, semantic, and syntactic) with the idea of extracting human and machine writing style patterns and being able to distinguish text from each other. For this reason, We applied a fine-tuning process using the four LLMs mentioned before: ERNIE[1], SpanBERT[2], ConvBERT[3], and XLNet[4] (using the Hugging Face library) for Subtask A Monolingual and Subtask B.

Starting with the data partition process, We used the same partition proposed by the organizers in the baseline code for both tasks. The training dataset was split into the train (80%) and validation (20%) for the fine-tuning process and the development dataset was used to measure the accuracy of each model with unknown data. Finally, the test dataset was only used to rank the models and verify the results.

Afterwards, in the fine-tuned process we tried with different hyperparameters on batch size (16, 32), learning rates (2e-5, 5e-5), random seed (0, 42), epochs (3, 5), and a weight decay of 0.01. Along with these params configurations, we used the Trainer, AutoModel, and AutoTokenizer classes from the Transformers. Each sequence was padded and truncated at 512 after tokenization due to the constraints of some of the models we used (most of them had a limit in the allowed length of the input sequence). These hyperparameters were chosen based on empirical experiments and hyperparameter tuning to achieve the best performance on our validation dataset. For the evaluation we computed macro-F1, micro-F1, and accuracy scores; being the last ones used by those organized to evaluate the final ranking.

Finally, in the test process, the output predictions for the model (logits) serve as an input for a Softmax function and then apply an argmax function in order to get the final prediction class.

5 Results

After the fine-tuning process using the training dataset, we measured the performance of each LLMs on the test set (development set provided) for Subtask A (Monolingual) and Subtask B (using the respective training and test data provided).

Table 1 shows the evaluation results for Subtask A (Monolingual) using the macro-F1, micro-F1, and accuracy measures (obtained from the score scripts provided for the organizers). For the Validation set, ERNIE's model outperforms the other LLMs across all metrics achieving 79.4 % in accuracy, but, ConvBERT and SpanBERT closely follow with 77.1% and 78.8% respectively.

For subtask A (Monolingual), We submitted our two best prediction results to the Codabench platform: ERNIE and ConvBERT LLMs. Table 3 shows the final ranking for this Subtask, we ranked place 87 out of 137 with an accuracy score of 71.5% (obtained by the ConvBERT LLM). The best team (safeai) obtained an accuracy score of 96.8% and the baseline (RoBERTa LLM) achieved 88.4%. Table 1 also shows the results evaluating these models in the test set (post-submission, using gold labels released by organizers); in this case, our best model was the ConvBERT with 77.6% in accuracy.

On the other hand, Table 2 shows the performance metrics obtained for SubTask B. In the Validation set, the SpanBERT model outperforms the other LLMs across all metrics achieving 66.8% of accuracy, 66.8% of micro-F1, and 63.4% of macro-F1 score. However, the ERNIE model closely follows with 65.4% accuracy; Then, we obtained the final predictions from the validation dataset using these fine-tuned trained models and uploaded to Codabench platform one submission based on the ConvBERT LLM results. Table 2 shows the final ranking for Subtask B, where we obtained place 67 out of 77 with an accuracy score of 38.2% (obtained by the ConvBERT LLM). In this case, the best team (tmarchitan) achieved an accuracy score of 86.9% and a baseline (RoBERTa LLM) of 74.6%. Finally, as in Subtask A, We re-evaluated these models on the test set released (post-submission), and, our best model was the XLNET with 65.2% in accuracy (second part in table 2).

On the other hand, figure 2 and figure 3 show the Confusion Matrix (CM) results for Subatsk A Monolingual and Subtask B, respectively. The CM for Subtask A across all models, presents a large confusion in classifying human text (True Positive vs False Positive) compared to the performance achieved for the machine-generated text (True Negative vs False Negative). For human text classification, the ConvBERT LLM was the best model

[1]https://huggingface.co/nghuyong/ernie-2.0-base-en
[2]https://huggingface.co/SpanBERT/spanbert-base-cased
[3]https://huggingface.co/YituTech/conv-bert-base
[4]https://huggingface.co/xlnet/xlnet-base-cased

SubTask A (Monolingual)					
Dataset	**Measure**	**Large Language Model**			
		ERNIE	**SpanBERT**	**ConvBERT**	**XLNET**
Validation Set	***macro-F1***	**0.789**	0.783	0.762	0.720
	micro-F1	**0.794**	0.788	0.771	0.733
	accuracy	**0.794**	0.788	0.771	0.733
Test Set*	***macro-F1***	0.701	0.760	**0.770**	0.758
	micro-F1	0.720	0.772	**0.776**	0.767
	accuracy	0.720	0.772	**0.776**	0.767

Table 1: Results obtained for each LLM on the Validation and Test set for Subtask A (monolingual).
* These results were obtained after the competition was finalized.

SubTask B					
Dataset	**Measure**	**Large Language Model**			
		ERNIE	**SpanBERT**	**ConvBERT**	**XLNET**
Validation Set	***macro-F1***	0.620	**0.634**	0.615	0.601
	micro-F1	0.654	**0.668**	0.640	0.634
	accuracy	0.654	**0.668**	0.640	0.634
Test Set*	***macro-F1***	0.578	0.518	0.603	**0.590**
	micro-F1	0.626	0.563	0.634	**0.652**
	accuracy	0.626	0.563	0.634	**0.652**

Table 2: Results obtained for each LLM on the Validation and Test set for Subtask B.
* These results were obtained after the competition was finalized.

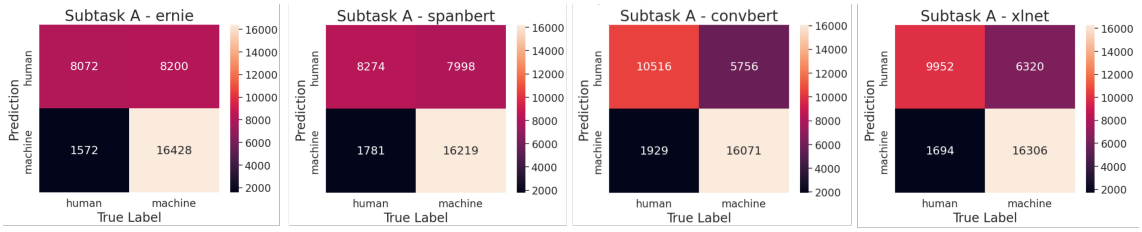

Figure 2: Subtask A Monolingual. Confusion Matrix results for each LLM applied on the test set (post-submission).

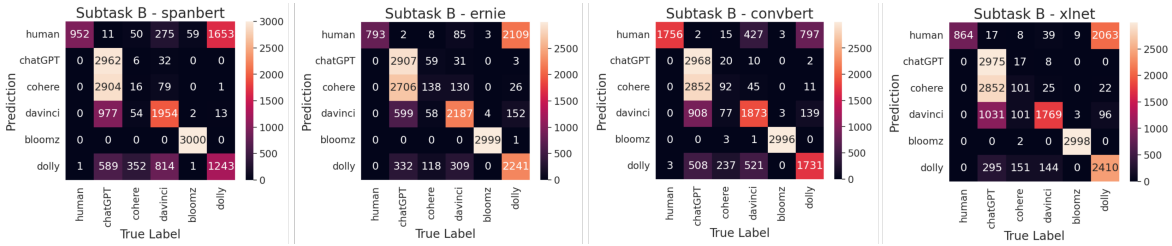

Figure 3: Subtask B. Confusion Matrix results for each LLM applied on the test set (post-submission).

getting 65% correct and 35% fail, meanwhile, the ERNIE LLM obtained a poor performance of 50% correct and 50% fail. Related to machine-generated text classification, all models performed similarly, obtaining less confusion: around 90% correct and 10% fail. Furthermore, CM for Subtask B in general struggles to classify human texts and presents a large confusion with dolly machine model across all LLM; the best performance was classifying chatGPT and bloomz text, getting around 98% correct in both, meanwhile, cohere and dolly machine obtained a poor classification performance.

Finally, We would like to mention that, due to some technical issues in our servers, We did not submit the models with the best model scores in the validation stages for both subtasks. For this reason,

Position	Team	Accurary
1	safeai	0.968
2	comp5	0.960
19	baseline	0.884
87	**iimasNLP (andric)**	**0.715**
137	saibewaraditya	0.231

Table 3: Final ranking per team in Subtask A (monolingual)

Position	Team	Accurary
1	tmarchitan	0.869
2	farawayxxc	0.843
24	baseline	0.746
67	**iimasNLP (andric)**	**0.382**
77	saibewaraditya	0.153

Table 4: Final ranking per team in Subtask B

We reported different scores in the final submission compared to our scores in the Test evaluation (post-submission, with gold labels).

6 Conclusion

We applied a fine-tuning process using four LLMs: ERNIE, SpanBERT, ConvBERT, and XLNet. In general, this LLM aims to extract lexical, semantic, and syntactic information from the text. We obtained comparable results with the baselines reported (initially), but, below compared to those in the first positions.

For future work, it could be interesting to prove more LLMs that focus on multilevel language and stylistic features; also apply a more robust finetuning process to evaluate more hyperparameters; and finally try a different approach based on text graph called Graph Neural Networks.

Acknowledgements

This paper has been supported by PAPIIT-UNAM projects IN104424, TA101722, and CONAHCYT CF-2023-G-64. The authors thank Ricardo Villareal and Rita Rodriguez for the technical support with computational resources and Roman Osorio for the student administration support. This work has the support of the CONAHCyT graduate scholarship program.

References

Harika Abburi, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. Generative ai text classification using ensemble llm approaches. *arXiv preprint arXiv:2309.07755*.

Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Piotr Przybyła, Nicolau Duran-Silva, and Santiago Egea-Gómez. 2023. I've seen things you machines wouldn't believe: Measuring content predictability to identify automatically-generated text. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023). CEUR Workshop Proceedings, CEUR-WS, Jaén, Spain*.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains. In *Procesamiento del Lenguaje Natural*, Jaén, Spain.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.