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

Aggarwal Pranjal, Sachdeva Deepanshu

University of Colorado Boulder

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

Abstract

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

1 Introduction

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

This Binary classification task has utmost utility in real world scenarios like - content moderation on social media platforms, fake news detection that can impact organizations financially and people emotionally, detecting spam messages in email or communication channels like Slack. Another application can be used in healthcare chatbots to make sure that a person is talking to a person as this kind of task needs human speciality. Product reviews classification - i.e., detecting whether an organization has human written reviews, or they had them generated through AI to rank their product higher up in the chain.

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

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

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

2 Background

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

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

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

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

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

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

Figure 1: AI Corpus Word Cloud
Key terms: Step, Part, Overall, S, One, Make

Figure 2: Human Corpus Word Cloud
Key terms: See, S, References, External, History

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

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

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

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

3 System Overview

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

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

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

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

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

4 Experimental Setup

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

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

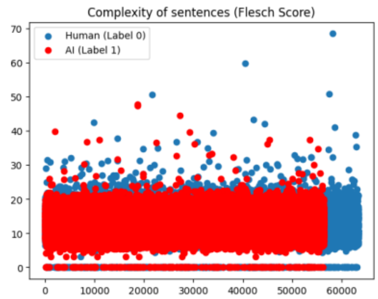

Figure 3: Complexity

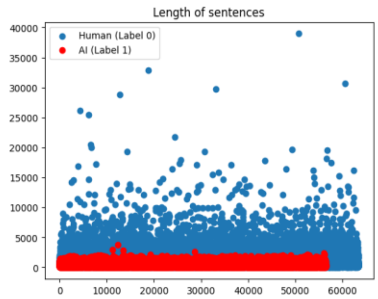

Figure 4: Length

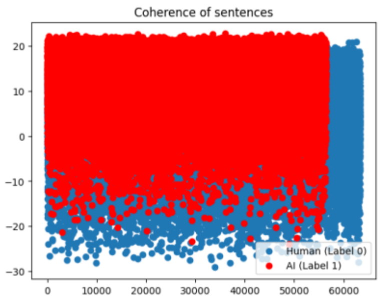

Figure 5: Coherence

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

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

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

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

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

5 Results

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

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

Table 1: Performance of different models

6 Conclusion

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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