Human vs Machine: An Automated Machine-Generated Text Detection Approach

Urwah Jwaid[⋆], **Rudra Roy**[†], **Pritam Pal**[‡], **Srijani Debnath**[⋄],
Dipankar Das[⋆] and **Sivaji Bandyopadhyay**[⋆]
[⋆]Jadavpur University, Kolkata, India
[†]Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India
[‡]RCC Institute of Information Technology, Kolkata, India
[⋄]Government College of Engineering and Leather Technology, Kolkata, India
{urwahjawaid, royrudra164, pritampal522, srijanidebnath2005, dipankar.dipnil2005, sivaji.cse.ju}@gmail.com

Abstract

With the advancement of natural language processing (NLP) and sophisticated Large Language Models (LLMs), distinguishing between human-written texts and machine-generated texts is quite difficult nowadays. This paper presents a systematic approach to classifying machine-generated text from human-written text with a combination of the transformer-based model and textual feature-based post-processing technique. We extracted five textual features: readability score, stop word score, spelling and grammatical error count, unique word score and human phrase count from both human-written and machine-generated texts separately and trained three machine learning models (SVM, Random Forest and XGBoost) with these scores. Along with exploring traditional machine-learning models, we explored the BiLSTM and transformer-based distilBERT models to enhance the classification performance. By training and evaluating with a large dataset containing both human-written and machine-generated text, our best-performing framework achieves an accuracy of 87.5%.

1 Introduction

In recent years the applications of Large Language Models (LLMs) such as ChatGPT[1], Microsoft Copilot[2], Google Gemini[3], etc. have been widely popular among high school students to aged professionals. People can perform any task using those LLM applications such as answering a question, summarising a chapter, checking grammatical errors and many other things which elevates their overall productivity and saves time.

However, a few users also misuse the power of LLM applications by doing homework assignments with ChatGPT or similar kinds of AI ChatBots, paraphrasing an article from one's original article etc. which are likely to be unethical practices. Moreover, with the increasing advancement of deep learning, generative AI and LLMs it is quite hard nowadays to distinguish whether a text is written by a human or an LLM-integrated ChatBot.

The main objective of this paper is to identify a text whether it is written by a human or it is generated by some LLM integrated ChatBot (machine-generated text). We employed different schemes to classify human-written text from machine-generated text starting from simple textual features to a deep learning model to a combination of deep learning and textual features. The main contributions in this paper can be summarized as follows:

- We extracted five textual features (readability score, unique word score, stop word score, spelling & grammatical error and human phrases count) from both human and machine-generated text.
- With these textual features we trained three machine learning models: SVM, Random Forest and XGBoost.
- Furthermore, we developed machine-generated text detection frameworks using the Bidirectional LSTM (Hochreiter and Schmidhuber, 1997) (BiLSTM) and transformer-based distilBERT (Sanh et al., 2019) model.
- Finally we developed a combined framework by performing some post-processing based on textual features on the top of the distilBERT model to get both the advantage from the distilBERT model and textual features.

2 Related Work

Although AI-generated text detection is a comparatively new domain in the field of NLP, a number of

[1] https://chatgpt.com/
[2] https://copilot.microsoft.com/
[3] https://gemini.google.com/

methods have been proposed by several researchers in the last couple of years.

A statistical method based machine-generated text detention tool named ‘GLTR’ was proposed by Gehrmann et al. (2019). Mindner et al. (2023) extracted several textual features and experimented with three machine-learning models to classify human-written text from ChatGPT-generated texts. Ippolito et al. (2020), Guo et al. (2023), Mitrović et al. (2023), Bhattacharjee et al. (2023) and Wang et al. (2024a) used transformer-based frameworks to detect AI-generated text and human-written text.

Bhattacharjee and Liu (2023) employed chatGPT with prompt engineering to identify human-written text and machine-generated text. Mitchell et al. (2023) proposed a zero-sort based machine-generated text detection approach (DetectGPT) using negative log probabilities. Bao et al. (2024) proposed another zero-sort based machine-generated text detection approach named ‘Fast-DetectGPT’ using conditional probability curvature. Their proposed approach outperforms the performance of the DetectGPT model and increases the machine-generated text detection process by 340 times compared to DetectGPT.

Hu et al. (2023) proposed an AI text detection tool ‘RADAR’ using adversarial learning taking paraphrasing of text into account. Wu et al. (2023) proposed a LLM generated text detection method using next-word probabilities.

Jawahar et al. (2020) presented a survey paper in which the authors critically explored state-of-the-art machine-generated text detention tools and performed in-depth error analysis. Weber-Wulff et al. (2023) explored 14 different types of AI-generated text detection tools including Turnitin and PlagiarismCheck type of widely used tools in their study to measure their reliability and accuracy. A benchmark consisting of a set of methods to assess the robustness of the existing AI-generated text detection models was proposed by Chakraborty et al. (2023). The authors also introduce the “AI detectability index (ADI)” to measure the AI detectability of different generations of LLM models.

3 Data

All the textual analysis and experiments were performed using the data collected from the “SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection”[4] (Wang et al., 2024b,c) shared-task. In this shared task, the organizers released both monolingual (English) and multilingual data, however, we had chosen only monolingual data for this current experiment.

This dataset contains approximately 1.2 lakh training examples, of which 63,351 were written by humans and 56,406 were generated by large language models such as ChatGPT, Cohere[5], BLOOMz[6] etc. Along with the training data, there were also development data and gold label testing data with a sample size of 5000 and 34,272 respectively. A distribution of all data is provided in Table 1. Also, a distribution of the number of words in the human text and machine-generated text is provided in Figure 1.

Data	Human	Machine
Training	63,351	56,406
Development	2,500	2,500
Testing	16,272	18,000
Total	82,123	76,906

Table 1: Distribution of Human-written and Machine-generated text.

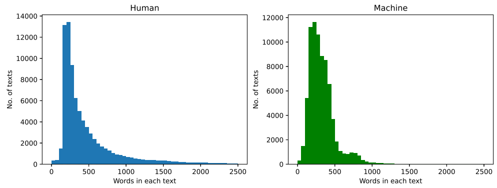

Figure 1: Distribution of the number of words in Human written text and machine-generated text.

4 Methodology

This section discusses the proposed methodologies for this work. Given a text T, our main aim is to determine whether it was written by a human (human-written) or generated by LLM models (machine-generated).

4.1 Text Preprocessing

Before diving into the text feature extraction and model training, the texts were first cleaned and preprocessed. After observing the texts in the dataset some preprocessing steps were applied such as:

[4] https://github.com/mbzuai-nlp/SemEval2024-task8

[5] https://cohere.com/

[6] https://www.bloomz.com/

1. Removal of escape characters such as '\n' (new-line character) and '\t' (tab character),
2. If any Unicode character is present in the text, then convert them into their corresponding ASCII values.

After preprocessing and cleaning, the texts were tokenized into a sequence of tokens or words $[k_1, k_2, k_3, ..., k_n]$. Note that, no other preprocessing such as stopword removal, stemming, lemmatization etc. was not performed.

4.2 Text analysis and Feature Extraction

Generally, human-written texts are more readable than human, i.e., human texts are easier to understand than machine-generated texts. Georgiou (2024) reported that the average Flesch-Kincaid Grade Level readability score is higher in machine-generated text than human-written texts. Therefore, we primarily calculated the Flesch-Kincaid Grade Level readability score for human and machine-generated text separately as it is an important feature to distinguish between human and machine-generated texts. Along with this we also calculated some additional textual features such as stop word score, unique word score, spelling and grammatical error etc.

4.2.1 Flesch-Kincaid Grade Level Readability Score

It is a measure of the readability of a piece of text proposed by Flesch (1948). It is commonly used in English language writing and education to assess the ease with which a reader can understand a given text. The higher the readability score the more text is difficult to understand. The formula for the Flesch-Kincaid Grade Level readability score (RS) is as follows:

$$RS = 0.39 \times \frac{\text{total words}}{\text{total sentences}} + 11.8 \times \frac{\text{total syllables}}{\text{total words}} - 15.59$$

4.2.2 Stop Word Score:

This metric was used to determine the distribution of stop-words such as 'a', 'an', 'the', etc in the human-written and machine-generated text. The stop word score (SWS) was calculated as the ratio of the number of stop words in a text to the total number of words in that text.

$$SWS = \frac{\text{total stop words}}{\text{total words}}$$

4.2.3 Unique Word Score:

The unique word score (UWS) was calculated by first removing the stop words and punctuations from a given piece of text and then dividing the number of unique words in the text by the total number of words.

$$UWS = \frac{|\{\text{all words}\} - \{\text{stop words}\}|}{\text{total words} - \text{total stop words}}$$

$\{\text{all words}\}$ and $\{\text{stop words}\}$ represent a set of words and a set of stop words in a text respectively. $\{\text{all words}\} - \{\text{stop words}\}$ represents the set difference and $|\{\text{all words}\} - \{\text{stop words}\}|$ defines the cardinality of the set.

4.2.4 Spelling and Grammatical Error

The spelling and grammatical errors (SGE) were checked using Python's 'language-tool-python'[7] package. This Python module counts the number of spelling and grammatical errors in a text and returns the number of error cases.

4.2.5 Human phrases count

Humans are more likely to use a few phrases such as 'However', 'Moreover', 'In addition', 'My pleasure', 'sorry to say' etc. than machines. Thus we count such phrases in a given piece of text and store the result. All the human phrases that we considered are provided in Appendix A.

The statistical analysis of the extracted features for human-written and machine-generated text for training data is provided in Table 2.

	Metric	Mean	Median	Mode	Std Dev
Human	RS	10.93	10.3	9.9	6.05
	SWS	0.37	0.38	0.33	0.07
	UWS	0.68	0.69	0.75	0.11
	SGE	95.23	48	25	172.05
	HPs	4.65	4	2	2.99
Machine	RS	11.34	11.3	11.8	5.94
	SWS	0.39	0.39	0.4	0.05
	UWS	0.63	0.64	0.67	0.13
	SGE	42.76	34	21	32.24
	HPs	2.85	2	2	2.03

Table 2: Statistical measure of human-written and machine-generated text in training data.

4.3 Framework Development

This section discusses the development of an AI-generated text classification framework. Several frameworks, from feature-based machine-learning

[7] https://pypi.org/project/language-tool-python/

models to deep learning models were developed to distinguish machine-generated text from human-written text.

4.3.1 Machine Learning Approach

We initially experimented with three machine learning models: Support Vector Machine (SVM), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) to recognize patterns that could classify machine-generated text from human-written text. In all three models, we passed the extracted five key features (readability score, stop word score, unique word score, spelling and grammatical error and human phrase count) as input to the machine-learning models.

To develop the SVM classifier we selected the radial basis function (RBF) kernel and to develop the RF classifier we selected the number of trees as 50 with the random state as 42 for all classifiers.

4.3.2 BiLSTM with GloVe

This approach involved the utilization of pre-trained GloVe (Pennington et al., 2014) embeddings to represent words in a continuous vector space, effectively capturing semantic relationships. The overall BiLSTM based framework is provided in Figure 2.

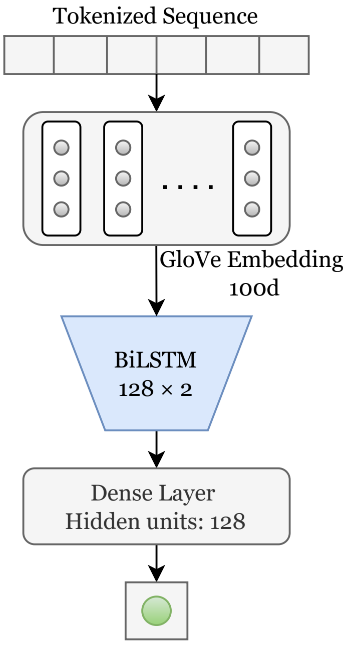

Figure 2: BiLSTM framework with GloVe embedding

The BiLSTM model framework consists of an embedding layer with pre-trained GloVe embedding with 100 dimensions, a BiLSTM layer with 128 hidden units to capture sequential dependencies and a dense layer of 128 hidden units.

$$Z_{dense} = ReLU(Z_{BiLSTM})$$

Where Z_{BiLSTM} and Z_{dense} represent the output of BiLSTM and dense layers respectively.

Classification: The final output layer consists of 1 hidden unit where the output of the dense layer (Z_{dense}) was passed as an input. The output layer used the sigmoid as its activation function.

$$P_{out} = sigmoid(Z_{dense})$$

$$\hat{Y}_* = \begin{cases} 0, P_{out} < 0.5 \\ 1, P_{out} \geq 0.5 \end{cases}$$

Where P_{out} represents the probability value and $\hat{Y}$ represents the predicted output class. '0' represents the human-written text and '1' represents machine-generated text.

4.3.3 distilBERT

The overall framework is depicted in Figure 3 where to develop the framework we used the transformer-based pre-trained distilBERT (Sanh et al., 2019) model. The distilBERT is a lightweight version of the BERT-base (Devlin et al., 2018) model which is 40% lighter and 60% faster than the BERT model, and retains 97% of the BERT model's language understanding (Sanh et al., 2019). In the distilBERT model, we passed the '`input ids`' which are basically the numeric representations of tokens generated by the distilBERT tokenizer and '`attention masks`'.

Next, the sequence output of the distilBERT model was passed to the '`GlobalMaxPooling`' layer and '`GlobalAvgPooling`' layer followed by a dropout of 0.2. The '`GlobalMaxPooling`' extracts the maximum value from each feature vector whereas the '`GlobalAvgPooling`' calculates the average value from each feature vector of the sequence output.

$$Z_{GlobalMaxPooling} = [Max(\hat{y_1}), Max(\hat{y_2}), Max(\hat{y_3}), ..., Max(\hat{y_l})]$$

and,

$$Z_{GlobalAvgPooling} = [Avg(\hat{y_1}), Avg(\hat{y_2}), Avg(\hat{y_3}), ..., Avg(\hat{y_l})]$$

Where $y_1, y_2, y_3, ..., y_l$ represent the feature vectors and l is the length of sequence output which is 768 for the distilBERT model.

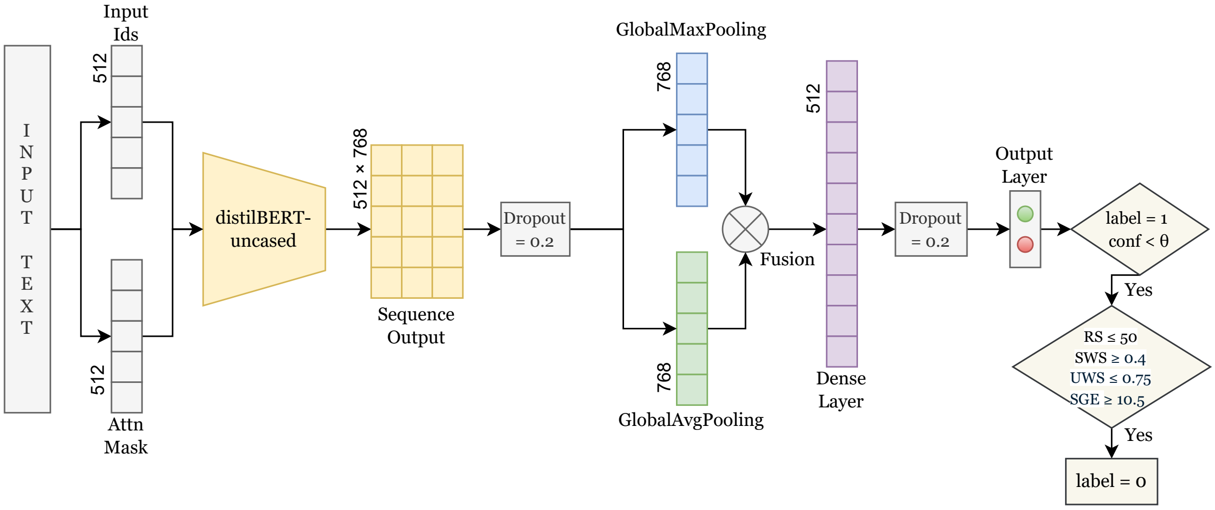

Figure 3: Proposed model framework with distilBERT and post-processing

Next, the output of the 'GlobalMaxPooling' and the 'GlobalAvgPooling' layers were concatenated to get a fused representation from two layers and then passed to a dense layer of 512 neurons with ReLU activation function.

$$Z_{dense} = ReLU(Z_{GlobalMaxPooling} \otimes Z_{GlobalAvgPooling})$$

$$Z_{dense} = Dropout(Z_{dense})$$

Where $\otimes$ represents the concatenation of layers and Z_{dense} represents the output of the dense layer.

Classification: The output of the dense layer followed by a dropout of 0.2 was passed to the final output layer with two neurons. The output layer used softmax as its activation function.

$$P_{out} = softmax(Z_{dense})$$

$$\hat{Y} = argmax(P_{out})$$

Where P_{out} represents the probability value and $\hat{Y}$ represents the predicted output class.

4.4 Training

To accomplish the training process, we used the same training and development split provided by "SemEval-2024 Task 8" organizers.

BiLSTM model: The proposed BiLSTM based framework was trained with a learning rate of 0.001 with 'BinaryCrosEntropy' loss function and batch size of 32. The optimizer was taken as Adam (Kingma and Ba, 2017) and the model was trained up to 5 epochs.

distilBERT model: In order to train the distilBERT based framework, we selected the 'SparseCategoricalCrossEntropy' loss function with a learning rate of 2e-5 and monitored the loss for the development data. The optimizer was taken as the Adam optimizer and the model was trained up to 3 epochs with a batch size of 32.

4.5 Post-processing

Upon training of the distilBERT model, a few post-processing steps were performed on the predicted output based on the prediction confidence and the extracted textual features as described in Section 4.2

If the predicted label was machine-generated text (output label = 1) and the model's prediction confidence was less than a value θ then only we check the threshold values. If the readability score (RS) was ≤ 50, the stop word score (SWS) was ≥ 0.4, the unique word score (UWS) was ≤ 0.75 and the spelling and grammar error (SGE) was ≥ 10.5 then we considered the text as a human-written text and changed the predicted label to human-written text (label = 0).

5 Experimental Setup and Result

5.1 Experimental Setup

All the experiments were accomplished using Python libraries such as 'nltk', 'scikit-learn' etc. The 'BiLSTM' and 'distilBERT' model was trained using the libraries of 'TensorFlow' and 'Keras' in the Kaggle environment with NVIDIA Tesla P-100 GPU and the machine learning (SVM, RF and XG-Boost) models were trained and evaluated using the 'scilit-learn' libraries. The Flesch-Kincaid readability score was calculated using the 'textstat'[8] library of Python.

To evaluate the performance of the machine-generated text classification framework, we calculated the accuracy and F1-score for the exact gold label test data provided by the organizers of "SemEval-2024 Task 8".

5.2 Result

We evaluated the results for the test data in the previously mentioned three machine learning models (SVM, RF, XGBoost), BiLSTM based framework and the following two schemes:

1. *Only distilBERT model*: Here we evaluated the results using the classification output of the distilBERT based framework. No post-processing or other steps were performed in this scheme.

2. *distilBERT with post-processing*: In this scheme, the post-processing steps were applied upon the output of the distilBERT classifier using the threshold values as mentioned in Section 4.5 based on the value of confidence score θ. We evaluated the performances on different θ values such as 0.5, 0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95 and 0.99.

The results of the three mentioned machine learning models, BiLSTM and distilBERT frameworks are provided in Table 3. The results after performing post-processing on the top of the distilBERT based framework are provided in Table 4.

	Accuracy	F1 Score
SVM	0.5229	0.5209
RF	0.5545	0.5493
XGBoost	0.5620	0.5532
BiLSTM	0.6845	0.6845
distilBERT	0.7742	0.7639

Table 3: Result of the proposed frameworks (without post-processing)

From Table 3 it is observed that among all the frameworks the distilBERT based framework performed well with accuracy and F1-score of 0.7742 and 0.7642. Moreover, when we checked the performance of machine-generated text and human-written texts separately, 93.5% of machine-generated texts and 60% of human-written texts were correctly identified by the distilBERT framework.

	Accuracy	F1 Score
conf < 0.5	0.7742	0.7639
conf < 0.6	0.7847	0.7759
conf < 0.7	0.8024	0.7958
conf < 0.75	0.8061	0.8027
conf < 0.8	0.8086	0.8027
conf < 0.85	0.8183	0.8133
conf < 0.9	0.8289	0.8248
conf < 0.95	0.8443	0.8413
conf < 0.99	0.8750	0.8736

Table 4: Performance after post-processing on the top of the distilBERT model on different confidence scores.

Regarding the BiLSTM framework, when we checked the performance of machine-generated text and human-written texts separately, 72% of human-written texts were correctly identified and only 65% of machine-generated texts were properly identified. It is also to be noted that the identification of human-written text in BiLSTM was improved by 16% compared to the identification of human-written text by the distilBERT. The confusion matrices for BiLSTM and distilBERT frameworks are provided in Figure 4.

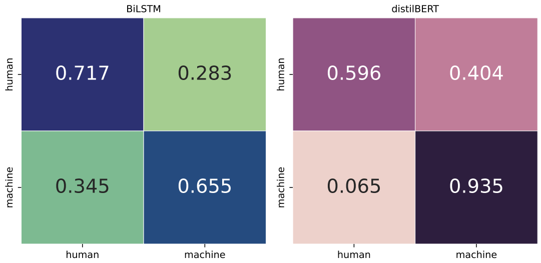

Figure 4: Confusion matrices for BiLSTM and distilBERT frameworks.

In contrast, The performances of SVM, RF and XGBoost machine learning models were not satisfactory with an F1 score of 0.5209, 0.5494 and 0.5532 respectively. One possible reason behind this is that the extracted five features (RS, SWC, UWS, SGE and HPs) by which we trained the machine learning models were not sufficient to accurately identify human-written and machine-generated texts. Thus, we got a result with low accuracy.

Result after post-processing: The results of distilBERT with post-processing are provided in Table 4 where we applied the post-processing based

[8] https://pypi.org/project/textstat/

on the confidence score value. From Table 4 it is observed that the accuracy and F1-score are increased as we increase the threshold value of the confidence score. At 'conf < 0.5' the performance was the same as the performance of the distilBERT framework and at 'conf < 0.99' we achieved the best result of 87.5% accuracy.

Upon applying post-processing it was also observed that the performance of human written text was notably increased. At 'conf < 0.95' the classification accuracy of human-written text was increased to 74% and at 'conf < 0.99' the classification accuracy of human-written texts was increased to 81%. However, the classification accuracy of machine-generated text was slightly decreased to 93.4% and 93.2% on the mentioned confidence scores respectively.

At 'conf < 0.9' and below (0.85, 0.8, 0.75 and so on) the classification accuracy of machine-generated text retains the same as the classification accuracy of distilBERT models with 93.5% accuracy. The confusion matrices for the best four post-processing schemes (conf < 0.99, conf < 0.95, conf < 0.9 and conf < 0.85) are provided in Figure 5.

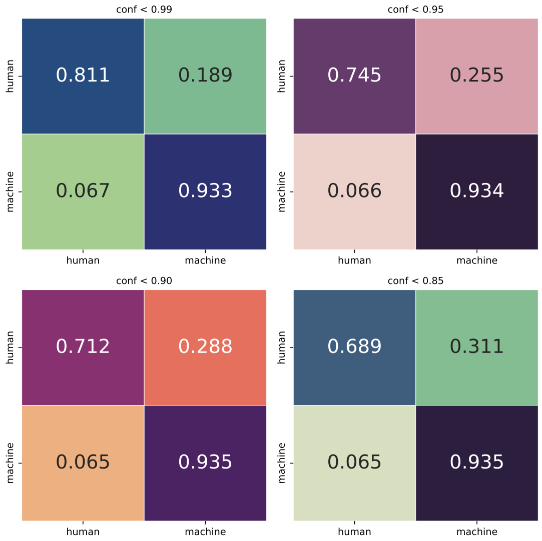

Figure 5: Confusion matrices for best four post-processing schemes

6 Conclusion

In this paper, we proposed machine learning and deep learning methods to classify the 'human-written' and 'machine-generated' texts leveraging textual features and transformer-based distilBERT model. Along with the distilBERT framework we also experimented with three machine-learning models and a deep-learning BiLSTM model.

Our proposed 'distilBERT with post-processing' scheme provides an accuracy of 87.5% which is a superior performance than the other proposed schemes.

Furthermore, it is observed that the distilBERT framework identifies machine-generated texts in more better way whereas the BiLSTM based frameworks identify human-written texts more accurately than distilBERT. However, the machine learning models didn't perform well and provided an unsatisfactory result.

In future, we'll evaluate our proposed 'distilBERT with post-processing' framework on other datasets and with other models such as BERT (Devlin et al., 2018) and RoBERTa (Conneau et al., 2019) to justify our claims more accurately and to validate the robustness of the proposed framework.

7 Limitations

Our proposed scheme also has several limitations. First, we considered only five textual features and didn't check the performance with other textual features such tone of a sentence, active/passive voice, sentiment polarity etc. Second, we used only BiLSTM and distilBERT models to develop our frameworks. The performance may be elevated by using other modes such as XLM-RoBERTa or BERT. Third, tokens were limited to 512 in the distilBERT model. The performance may be improved if long sequences of tokens can provided as input to the model. However, using a long sequence of tokens requires more computational cost and time. And lastly, all the experiments were performed only on SemEval-2024 Task 8 data. It is necessary to evaluate the performance of the proposed methods on other datasets to verify their robustness.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Amrita Bhattacharjee and Huan Liu. 2023. Fighting fire with fire: Can chatgpt detect ai-generated text?

Megha Chakraborty, S.M Towhidul Islam Tonmoy, S M Mehedi Zaman, Shreya Gautam, Tanay Kumar, Krish Sharma, Niyar Barman, Chandan Gupta, Vinija Jain, Aman Chadha, Amit Sheth, and Amitava Das. 2023. Counter Turing test (CT2): AI-generated text detection is not as easy as you may think - introducing AI detectability index (ADI). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2206–2239, Singapore. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Georgios P. Georgiou. 2024. Differentiating between human-written and ai-generated texts using linguistic features automatically extracted from an online computational tool.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning.

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

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. 2023. *Classification of Human- and AI-Generated Texts: Investigating Features for ChatGPT*, page 152–170. Springer Nature Singapore.

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

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

Hao Wang, Jianwei Li, and Zhengyu Li. 2024a. Ai-generated text detection and classification based on bert deep learning algorithm.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024c. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Debora Weber-Wulff, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltýnek, Jean Guerrero-Dib, Olumide Popoola, Petr Šigut, and Lorna Waddington. 2023. Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*, 19(1).

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. LLMDet: A third party large language models generated text detection tool. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2113–2133, Singapore. Association for Computational Linguistics.

A Appendix: Human Phrases

1) ‘because of’,
2) ‘besides’,
3) ‘consequently’,
4) ‘etc’,
5) ‘furthermore’,
6) ‘have you’,
7) ‘honestly’,
8) ‘however’,
9) ‘just’,
10) ‘just imagine’,
11) ‘I’,
12) ‘i believe’,
13) ‘i think’,
14) ‘imagine’,
15) ‘in contrast’,
16) ‘in the same way’,
17) ‘let’,
18) ‘likewise’,
19) ‘meanwhile’,
20) ‘mine’,
21) ‘moreover’,
22) ‘my’,
23) ‘my pleasure’,
24) ‘nevertheless’,
25) ‘nonetheless’,
26) ‘on the other hand’,
27) ‘probably’,
28) ‘similarly’,
29) ‘sorry’,
30) ‘sorry to say’,
31) ‘suppose’,
32) ‘thank you’,
33) ‘think about it’,
34) ‘think about that’,
35) ‘thus’,
36) ‘what do you’,
37) ‘when do you’,
38) ‘whatever’,
39) ‘you’,
40) ‘your’,
41) ‘yourself’