Sharif-MGTD at SemEval-2024 Task 8: A Transformer-Based Approach to Detect Machine Generated Text

Seyedeh Fatemeh Ebrahimi ♣, **Karim Akhavan Azari**♣, **Amirmasoud Iravani**★
Arian Qazvini♢, **Pouya Sadeghi**†, **Zeinab Sadat Taghavi**♣, **Hossein Sameti**♣
Ferdowsi University of Mashhad, Mashhad, Iran★
Amirkabir University of Technology, Tehran, Iran♢
University of Tehran, Tehran, Iran†
Sharif University of Technology, Tehran, Iran♣

{sfati.ebrahimi, karim.akhavan, zeinabtaghavi, sameti}@sharif.edu
a.iravani@mail.um.ac.ir
a.qazvini@aut.ac.ir
pouya.sadeghi@ut.ac.ir

Abstract

Detecting Machine-Generated Text (MGT) has emerged as a significant area of study within Natural Language Processing. While language models generate text, they often leave discernible traces, which can be scrutinized using either traditional feature-based methods or more advanced neural language models. In this research, we explore the effectiveness of fine-tuning a RoBERTa-base transformer, a powerful neural architecture, to address MGT detection as a binary classification task. Focusing specifically on Subtask A (Monolingual - English) within the SemEval-2024 competition framework[1], our proposed system achieves an accuracy of 78.9% on the test dataset, positioning us at 57th among participants. Our study addresses this challenge while considering the limited hardware resources, resulting in a system that excels at identifying human-written texts but encounters challenges in accurately discerning MGTs.

1 Introduction

Recent advancements in large language models (LLMs) have endowed them with an impressive capability to generate written text that closely resembles human writing (Adelani et al., 2019; Radford et al., 2019). However, this technological progress brings along significant challenges, as the proliferation MGT poses various threats in digital environments. MGTs have been implicated in spreading misinformation in online reviews, eroding public trust in political or commercial campaigns, and even facilitating academic fraud (Crothers et al., 2022; Song et al., 2015; Tang et al., 2023). The identification of MGT remains a pressing concern, as distinguishing between human-written and machine-generated content is often challenging for humans. Consequently, there is a growing imperative to develop automatic systems capable of discerning MGT (Mitchell et al., 2023). In this study, we address this challenge within the English language context using the dataset provided by Wang et al. (2023).

As highlighted in Wang et al. (2024b) overview paper on the task, recent approaches to MGT detection predominantly employ binary classification methods. Existing literature highlights the superior performance of transformer-based methods over alternative approaches Wang et al. (2024a). However, a significant challenge in utilizing these models lies in the requirement for GPU hardware and computational resources. Our study aims to address this challenge within the constraints of limited hardware capacity. Keeping this in mind, we propose a system that leverages fine-tuning of the RoBERTa transformer model (Liu et al., 2019) to automatically classify input text as either human-written or machine-generated. Our system architecture involves augmenting the RoBERTa-base model with a Classifier Head. The Embeddings component facilitates contextual understanding of texts, while the Encoder component processes input texts in parallel, and the Classifier Head performs binary classification by linearly outputting a single value.

Our proposed system achieves an accuracy of 78.9% on the test data, surpassing the average results provided by the task's baseline and ranking 57th among 140 participants. The area under the

[1] https://semeval.github.io/SemEval2024/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 565–572

ROC curve (AUC) metric is measured at 0.69. While the ROC curve analysis demonstrates our model's capability to classify substantial portions of positive cases, its proximity to the diagonal line indicates room for further improvement. Notably, our primary challenge stemmed from computational constraints, which limited our ability to implement larger token sizes or batch sizes. Further discussions reveal that our system encounters difficulties in accurately detecting MGTs. To facilitate reproducibility and further research in this area, the code for our system is available on GitHub[2].

2 Background

2.1 Dataset Overview

SemEval-2024 Task 8 (Wang et al., 2024b) comprises three subtasks, with our investigation centering on Subtask A: binary classification of human-written versus MGT. Specifically, we concentrated our efforts on analyzing English monolingual data, as outlined dataset is provided by Wang et al. (2023).

Subtask A encompasses a dataset consisting of 119,757 training examples and 5,000 development examples, all presented in JSON format. Each data instance includes the following attributes:

- *id*: An identifier number for the example.
- *label*: A binary label indicating whether the text is human-written (0) or machine-generated (1).
- *text*: The actual textual content.
- *model*: The AI machine responsible for generating the text.
- *source*: The web domain from which the text originates.

2.2 Related Work

MGT detection is feasible through both traditional feature-based methods and neural language models. Fröhling and Zubiaga (2021) and Nguyen-Son et al. (2018) discussed how feature-based methods leverage statistical techniques. These methods primarily utilize frequency features such as TF-IDF, linguistic cues, and text style (Fröhling and Zubiaga, 2021). However, feature-based methods have limitations, as different samplings in language models can lead to varied generated outputs (Holtzman et al., 2019). In contrast, methods that harness neural language models, particularly those employing transformer models, have shown high effectiveness (Crothers et al., 2022). Neural language model methods often involve zero-shot classification or fine-tuning pre-trained language models (Sadasivan et al., 2023). Grover by Zellers et al. (2019), RankGen by Krishna et al. (2022), and DetectGPT (Mitchell et al., 2023) are prominent examples of zero-shot methods. However, these methods may be misleading at times and exhibit limited performance in out-of-domain tasks (Crothers et al., 2022; Wang et al., 2023).

Bakhtin et al. (2019) demonstrated outstanding performance in MGT detection by harnessing bidirectional transformers. Additionally, Solaiman et al. (2019) highlight that the zero-shot methods often fall short compared to a simple TF-IDF baseline when detecting texts from diverse domains. He argues that bidirectional transformers offer significant advantages for MGT detection, advocating for the fine-tuning of these models as a superior alternative to zero-shot methods. In this regard, Rodriguez et al. (2022) observed a significant enhancement in performance of cross-domain MGT detection by fine-tuning the RoBERTa detector.

Jawahar et al. (2020) conducted a comprehensive survey of various approaches to developing MGT detectors. Their findings suggest that fine-tuning the RoBERTa detector consistently delivers robust performance across diverse MGT detection tasks, surpassing the efficacy of traditional machine learning models and neural networks. Additionally, Crothers et al. (2022) reported a notable trend towards the increased utilization of bidirectional transformer architectures, particularly RoBERTa, in MGT detection tasks. Lastly, Wang et al. (2024a) conducted a comprehensive benchmark of supervised methods on M4 dataset. Their findings revealed that transformer models such as RoBERTa and XLM-R exhibited superior performance across all tests, respectively achieving 99.26% and 96.31% accuracy in MGT binary classification.

While this review does not provide a comprehensive examination of all aspects of

[2] https://github.com/Sharif-SLPL/Sharif-MGTD

MGT detection, prior research underscores the prevalence of transformer-base methods, like RoBERTa and XLM-R, in comparison to alternative approaches, especially in supervised tasks. Moreover, the superiority of RoBERTa over other models is evident. A significant challenge for studies utilizing pre-trained transformer models lies in the necessity for robust GPU hardware and computational resources.

3 System Overview

This section presents an overview of our system's architecture, highlighting implementation details and challenges. Drawing on the preceding works discussed above, which showed the efficacy of fine-tuning RoBERTa models, our system aims to attain peak performance in MGT detection while optimizing configurations for limited hardware resources.

The decision to employ the transformer architecture for detecting synthetic texts is motivated by its capacity to capture intricate dependencies within textual data. This choice seems logical considering that such texts often exhibit semantic features that can be harnessed for fact-checking, cohesion, coherence, and other properties that may unveil their origin (Raj et al. (2020). In contrast to traditional architectures, the transformer model overcomes the constraints of fixed window sizes or sequential processing, enabling it to utilize contextual information from the entire input sequence. Additionally, the self-attention mechanism empowers the model to selectively focus on pertinent segments of the input, rendering it highly effective for tasks necessitating long-range dependencies and contextual comprehension.

As for RoBERTa, it is specifically chosen for its extensive training duration, broader dataset coverage, ability to handle longer sequences, and focus on Natural Language Understanding tasks, making it more suitable than other BERT-based models. Additionally, a wealth of research, such as the recent study of Wang et al. (2024a), has further highlighted the inherent potential of RoBERTa for this specific task.

3.1 Core Algorithms and System Architecture

At the core of our system lies the concept of binary classification, distinguishing input texts as either machine-generated or human-written through fine-tuning a pre-trained RoBERTa transformer (Liu et al., 2019). Our system architecture entails augmenting the RoBERTa-base model with a Classifier Head. The RoBERTa model's Embeddings component incorporates a 768-dimensional embedding matrix, alongside position and token type embeddings, enhancing contextual understanding. The Encoding component features a 12-layer RoBERTaEncoder, each layer employing a multi-head self-attention mechanism. This facilitates simultaneous attention to different parts of the input text, crucial for analyzing textual similarities. Intermediate sub-layers utilize a fully connected feed-forward network with GELU activation, followed by an output sub-layer for feature transformation and normalization.

The Classifier Head, integrated into the Encoder for sequence classification, comprises a linear layer with 768 input features and a dropout layer to mitigate over-fitting. The final output is generated through an additional linear layer with a solitary output neuron, making it conducive to binary classification tasks. In essence, the primary model processes input data, with the Classifier Head making predictions. When viewed as a regression task, the Classifier produces a linear output tailored for a singular class, providing a probabilistic value. Implementation of the system is facilitated using PyTorch, incorporating specific parameters such as the AdamW optimizer (Radford and Narasimhan, 2018) and the CrossEntropyLoss function (Hui and Belkin, 2020). AdamW, renowned for training deep neural networks, integrates weight decay to mitigate over-fitting. The Cross Entropy Loss function, commonly employed in multi-class classification scenarios, combines softmax activation with negative log-likelihood loss. The training process involves iterating through the entire dataset for two epochs, with early stopping mechanisms in place to terminate training at the optimal point.

3.2 System Challenges

While larger machine-generated documents often exhibit more discernible patterns and clues, such as incoherence or repetition, they also entail substantial computational costs. Our primary challenge lay in efficiently processing these large documents using cost-effective computing systems. To mitigate this challenge, we explored strategies such as reducing token size and batch size. However, these adjustments necessitate trade-offs, potentially leading to reduced accuracy or

increased processing time.

Our system was trained using a token size of 512, but optimal performance could potentially be achieved with larger token sizes, such as 1024 or 2048, given sufficient computing resources.

4 Experimental Setup

4.1 Dataset

Table 1 presents detailed statistics on the dataset used for each class.

Class/Split	Train	Test	Development
Human-Written Text	57075	6276	2500
Machine-Generated Text	50706	5700	2500

Table 1: Dataset Statistics

As shown in Table 1, nearly 90% of the dataset is dedicated to training, while the remainder is used for evaluation. To enhance model performance, we utilized the entire development dataset for model selection, compensating for the scarcity of training data.

4.2 Pre-processing and Hyper-Parameter Tuning

Input texts are tokenized using the RoBERTa tokenizer before processing, both during training and inference. Our hyper-parameter tuning process involved a comprehensive exploration across various parameter ranges. Specifically, we conducted experiments with learning rates ranging from 0.0001 to 0.00004, dropout rates spanning from 0.1 to 0.3, batch sizes varying between 4 and 16, and token sizes ranging from 64 to 1024. Through experimentation and analysis, we determined the optimal hyper-parameter settings, which are as follows: a learning rate of 0.00004, a dropout rate of 0.1, a token size of 512, a batch size of 10, and a weight decay of 0.01. Further details are given in Appendix A.

As illustrated in Appendix A, the number of training instances is correlated with the input token size and may influence the model accuracy. Given the length of input texts, a suitable token size is essential to capture all tokens adequately. However, computational costs associated with larger token sizes present a significant challenge during model training. Consequently, we selected 512 as the optimal token size. Truncation was employed during tokenization to accommodate the chosen token size, ensuring efficient model training without compromising data representativeness.

4.3 Training Procedure

For training the model, we utilized the Task dataset Wang et al. (2023), which underwent preprocessing by tokenizing the text into sub-word units and padding sequences to a fixed length. CrossEntropyLoss was employed as the loss function. The implementation also involved the AdamW optimizer, known for its effectiveness in training deep neural networks and its incorporation of weight decay to address over-fitting. The Adam optimizer was utilized with a learning rate of 4e-05. During training, the loss was monitored on a held-out validation set, and early stopping was applied to prevent over-fitting. Early stopping was implemented with the condition that the training loss reached a specific threshold (0.35 in this case), typically occurring around the third epoch. Therefore, if there was no improvement in the validation loss for a certain number of epochs, training was halted to prevent over-fitting of the model.

4.4 Evaluation Measures

The evaluation of our model involves calculating its accuracy in predicting whether a text is human-written or machine-generated. Accuracy, a fundamental metric in classification tasks, assesses the overall correctness of predictions and is calculated as:

$$Accuracy = \frac{ni}{N} \times 100 \quad (1)$$

where n_i represents the number of correctly classified instances, and N is the total number of instances.

5 Results

Using the official accuracy metric of SemEval-2024 Task 8 (Wang et al., 2024b), our system achieved the following accuracy scores on different data splits:

Language /Split	Devset	Testset
English	74.8%	78.9%

Table 2: Accuracy Metric

A direct comparison of our results with prior works is challenging due to the unique nature of

our research. To the best of researchers' knowledge, the most comprehensive benchmark on supervised MGT detection is presented by Wang et al. (2024a) using the M4 dataset and employing RoBERTa, XLM-R, GLTR-LR, GLTRSVM, Stylistic-SVM, and NELA-SVM. However, our primary objective was to determine strategies for addressing limited hardware resources as discussed in Appendix A.

As a contribution to this field, through repeated experiments, we identified that among hyperparameters, token size plays a slightly more significant role in model accuracy. While the system's accuracy is influenced by increasing the token size, drawing meaningful scientific conclusions necessitates further controlled experiments. Additionally, the expansion of token size is restricted by hardware limitations, requiring a detailed investigation with robust computational resources like GPU or TPU. Considering the constraints of Google Colab's[3] Free runtimes, we opted for a token size of 512 as a balance between hardware limitations and time constraints. Consequently, based on the official accuracy metric of SemEval2024 Task 8 (Wang et al., 2024b), our system achieved the following accuracy scores on various data splits:

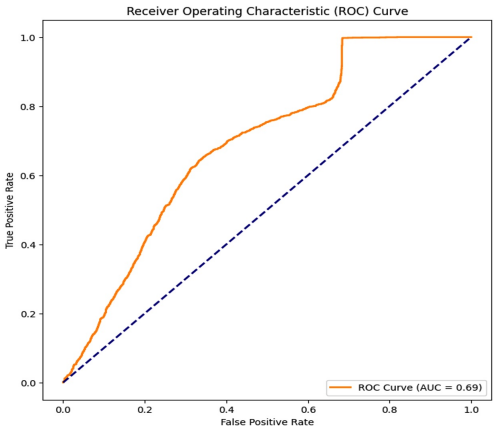

Figure 1: The ROC Curve Plot

The evaluation of our model also included analysis of the Area Under the Curve (AUC), a crucial metric that reflects the discriminative power of a binary classification model. Our fine-tuned RoBERTa model demonstrated an AUC of 0.69, suggesting its ability to effectively distinguish between positive and negative instances. Figure 1 illustrates the Receiver Operating Characteristic

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

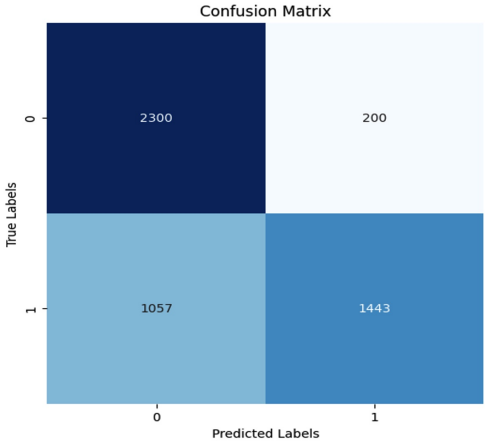

Figure 2: The Confusion Matrix Plot

(ROC) Curve, depicting the model's capability to accurately classify a significant proportion of positive cases. However, the proximity of the curve to the diagonal line suggests opportunities for further enhancement.

Interestingly, analysis of the confusion matrix, as depicted in Figure 2, revealed notable patterns in our model's classification tendencies. While our system effectively identified human-written documents with low False Positives, it exhibited difficulties in correctly identifying MGTs. This observation suggests potential areas for refinement, particularly in enhancing the model's ability to detect subtle cues and characteristics unique to machine-generated content.

Overall, our study contributes to the ongoing efforts in the field of NLP by showcasing the effectiveness of fine-tuned transformer models, particularly RoBERTa, in MGT detection tasks. Moving forward, future research directions could explore novel approaches to mitigate computational costs and further improve the performance of MGT detection systems, ultimately advancing the capabilities of NLU models in real-world applications.

6 Conclusion

In summary, our study focused on fine-tuning a RoBERTa-base transformer model for binary classification, specifically in distinguishing human-written from MGT. While our system showed promise in identifying human-written text, it faced challenges with accurately classifying machine-generated content. As discussed in Appendices A and B, we recommend exploring larger token sizes to improve model performance, albeit with

awareness of computational costs. Additionally, we advocate for the development of low-cost algorithms capable of efficient processing across hardware platforms. Our findings contribute to advancing MGT detection, with implications for combating misinformation and enhancing cybersecurity in the digital age.

Acknowledgments

We appreciate the Speech and Language Processing Laboratory at Sharif University of Technology[4] for providing us with this opportunity for collaborative work.

References

David Ifeoluwa Adelani, Hao Thi Mai, Fuming Fang, Huy Hoang Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *International Conference on Advanced Information Networking and Applications*.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *ArXiv*, abs/1906.03351.

Evan Crothers, Nathalie Japkowicz, and Herna L. Viktor. 2022. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751.

Like Hui and Mikhail Belkin. 2020. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *ArXiv*, abs/2006.07322.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. In *International Conference on Computational Linguistics*.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. *ArXiv*, abs/2205.09726.

[4] https://github.com/Sharif-SLPL

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

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*.

Hoang-Quoc Nguyen-Son, Ngoc-Dung T. Tieu, Huy Hoang Nguyen, Junichi Yamagishi, and Isao Echizen. 2018. Identifying computer-generated text using statistical analysis.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Mayank Raj, Ajay Jaiswal, Rohit R.R, Ankita Gupta, Sudeep Kumar Sahoo, Vertika Srivastava, and Yeon Hyang Kim. 2020. Solomon at SemEval-2020 task 11: Ensemble architecture for fine-tuned propaganda detection in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1802–1807, Barcelona (online). International Committee for Computational Linguistics.

Juan Diego Rodriguez, Todd Hay, David Gros, Zain Shamsi, and R. Srinivasan. 2022. Cross-domain detection of gpt-2-generated technical text. In *North American Chapter of the Association for Computational Linguistics*.

Vinu Sankar Sadasivan, Aounon Kumar, S. Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *ArXiv*, abs/2303.11156.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *ArXiv*, abs/1908.09203.

Jonghyuk Song, Sangho Lee, and Jong Kim. 2015. Crowdtarget: Target-based detection of crowdturfing in online social networks. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *ArXiv*, abs/2303.07205.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar

Habash, Iryna Gurevych, and Preslav Nakov. 2024a. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. *ArXiv*, abs/2402.11175.

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, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *ArXiv*, abs/2305.14902.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *ArXiv*, abs/1905.12616.

A Hyper-Parameter Tuning

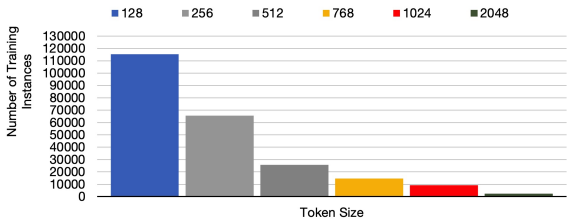

Figure 3: Number of Training Instances by Token Size

To determine the appropriate settings for hyper-parameters, we utilized Google Colab's free GPU runtime. Free Colab users have access to GPU and TPU runtimes without charge for a maximum of 12 hours. The GPU runtime includes an NVIDIA Tesla K80 with 12GB of VRAM. [Date: 5 Dec 2023]. We were unable to use premium runtime accounts due to financial issues arising from Iran sanctions. Therefore, we couldn't change our model's token size to larger than 512 due to the 12-hour time limit in free Colab. To understand the impact of increasing token size, we aimed to experiment on a local laptop GPU.

During the experiments aimed at finding the proper token size, we encountered the "CUDA error: device-side assert triggered" frequently, which was resolved by restarting the session. Our experiments were conducted using an RTX 2060 mobile with 6 GB of VRAM. Throughout all experiments, we maintained fixed parameters, including Number of Epochs = 3, Train Split = 0.7, and Learning Rate = 4e-05. Increasing the Max Length from 512 to 1024 in this experimental setup resulted in an improvement in Test Accuracy by at least 2%. However, this enhancement came at the cost of a nearly 15-fold decrease in training speed, making it challenging to implement on limited hardware. Additionally, this requires plenty of controlled experiments by researchers to shed light on finding the proper hyper-parameters.

B Detect-GPT as a Zero-Shot Method

In our pursuit of effective MGT detection, we also experimented with Mitchell et al. (2023)Detect-GPT model, a zero-shot approach utilizing probability curvature analysis. Training the model resulted in an accuracy rate of 60%, and when applied to a test dataset of approximately 1500 samples, it achieved a remarkable accuracy of approximately 84%. We conducted a comprehensive analysis by implementing 10 perturbations for each dataset. To address data and mask filling tasks, we employed the T5 small model, leveraging its robust capabilities. Furthermore, to accurately assess the log likelihood, we utilized the GPT-2 model, ensuring precise calculations and reliable results.This method surpassed alternative text detection methodologies, demonstrating superior accuracy and reliability in identifying MGT. Notably, the inclusion of threshold configuration added granularity to the experiment, enabling fine-tuning of detection sensitivity across varying threshold settings.