MAGRET: Machine-generated Text Detection with Rewritten Texts

Yifei Huang[a,b], **Jingxin Cao**[a,b*], **Hanyu Luo**[a,b], **Xin Guan**[a,b], **Bo Liu**[c]
[a]School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China
[b]Key Laboratory of Computer Network and Information of Ministry of Education of China,
Nanjing 211189, China
[c]School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
{huang_yifei, jx.cao, luohanyu, xin_guan, bliu}@seu.edu.cn

Abstract

With the quick advancement in text generation ability of Large Language Model(LLM), concerns about the misuse of machine-generated text(MGT) have grown, raising potential violations of legal and ethical standards. Some existing studies concentrate on detecting machine-generated text in open-source models using in-model features, but their performance on closed-source large models is limited. This limitation occurs because, in the closed-source model detection, the only reference that can be obtained is the texts, which may differ significantly due to random sampling. In this paper, we demonstrate that texts generated by the same model can align both semantically and statistically under similar prompts, facilitating effective detection and traceability. Specifically, we fine-tune a BERT encoder through contrastive learning to achieve semantic alignment in randomly generated texts from the same model. Then, we propose a method called **Ma**chine-**G**enerated Text Detection with **Re**written **T**exts, which designed several prompt refactoring methods and used them to request rewritten text from LLMs. Semantic and statistical relationships between rewritten and original texts provide a basis for detection and traceability. Finally, we expanded the text dataset with multi-parameter random sampling and verified the performance of MAGRET on three text-generated datasets. Experimental results show that previous methods struggle with closed-source model detection, while our approach significantly outperforms baseline methods in this regard. It also shows MAGRET's stable performance in detection and tracing tasks across various randomly sampled texts.

1 Introduction

With the emergence of big language models(Achiam et al., 2023; Touvron et al., 2023; Du et al., 2021), a large number of machine texts are generated and enter the human community. LLMs have been demonstrated to possess the ability to participate in exams(Chang et al., 2023) and mimic human behavior. The abuse of machine-generated text may violate legal and ethical standards(Tamkin et al., 2021), while the cost of manually determining whether text is machine-generated is prohibitively high. Therefore, an efficient and accurate machine-generated text detector is an important tool. Furthermore, tracing the origins of text represents an innovative and far-reaching field, offering more precise labeling for textual content.

Existing research primarily focuses on open-source models(Wang et al., 2023; Li et al., 2023), predicting outcomes based on extracted log-probability features. In practice, closed-source models are typically larger, perform better, and dominate the market. The rapid update cycle of closed-source models, exemplified by OpenAI's release of two models and multiple features in 2024 (OpenAI, 2023), exacerbates this issue. Consequently, current methods fall short in meeting the demands for detection and tracing of closed-source models.

One of the biggest difficulties in detecting and tracing closed-source models generated texts is that there is no reference other than re-requested text that can be used to make prediction, and some of the existing research ideas make Out-of-Distribution prediction by obtaining the hidden-layer weights of the open-source LLMs (Mireshghallah et al., 2024). However, the performance of this method's degrades as the number of black-box models increases.

The text generated by a big language model is affected by multiple control variables, such as top-p, top-k, and temperature of the pre-trained model. subtle differences in the prompt can also make huge changes in the generated content of the big language model. The detection model needs to distinguish the generated text of different models

[*]Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 8336–8346
January 19–24, 2025. ©2025 Association for Computational Linguistics

and exclude the interference of different generation parameters under the same model. Our study found that although these parameters can drastically change the content of the generated text, different models can still be distinguished by semantic and statistical analysis.

We propose the MAGRET model, which predicts whether a text is machine-generated and identifies the LLMs responsible by obtaining duplicated texts from LLMs through rewriting and continuation requests. We demonstrate that even with random sampling, the rewritten texts maintain semantic and statistical similarity to the original texts. For semantic alignment, we utilize BERT, and we identify several applicable similarity algorithms for statistical analysis. Our detection model requires only machine-generated natural language text, whether it is an open-source or close-source model. We tested MAGRET on three major language generation tasks, and MAGRET is quite effective under different top-p, temperature parameter and Out-of-Distribution. Compared to previous research, MAGRET has better detection and traceability results for closed-source models, providing a sustainable, generalized, and robust method for detecting and tracing content generated by LLMs.

In summary, our contributions are as follows.

- To the best of our knowledge, MAGRET is the first model to detect machine-generated text using complete rewritten sentence, without requiring the model to be open-source.
- MAGRET can detect generated text across various random sampling parameters, thereby broadening the detection scope and showing distinct features to differentiate high-random-sampling machine-generated text from human-written content.
- We expanded the dataset under text generation tasks such as Writing, QA, and Review, incorporating binary, multiclass, different random sampling parameters and out-of-distribution experiments. MAGRET consistently exhibits advanced detection performance.

2 Related Works

Binary Detection Traditionally, MGT detection has been framed as a binary classification problem, distinguishing between human-written and machine-generated text (Gehrmann et al., 2019; Ippolito et al., 2019). Supervised approaches in this domain rely on annotated datasets to train classifiers (Wang et al., 2024; Uchendu et al., 2021; Zhong et al., 2020; Liu et al., 2022). Recent studies by (Guo et al., 2023; Hu et al., 2023; Xiong et al., 2024) have further explored and refined these supervised methods, underscoring the continuing relevance of this approach.

Multi-Class Detection As the field progresses, there is growing interest in more fine-grained classification that not only identifies whether a text is machine-generated or human-written but also determines its specific source (i.e., which Large Language Model generated it). This multi-class classification problem shares similarities with authorship attribution (Uchendu et al., 2020; Munir et al., 2021). (Venkatraman et al., 2023) investigated whether the principle of humans' tendency to spread information evenly could help capture unique signatures of LLMs and human authors. The M4GT-Bench (Wang et al., 2024) focused on black-box modeling and multilingual text detection. (Shi et al., 2024) proposed an approach similar to ours, using resampling from the model to enhance prediction, but their method differs in not utilizing the full text, necessitating multiple sampling.

Linguistic Pattern-Based Detection Another significant line of research explores linguistic patterns for automatic machine-writing detection. This approach has evolved through various methods, including: N-gram frequencies (Badaskar et al., 2008) Entropy analysis (Lavergne et al., 2008; Gehrmann et al., 2019) Perplexity measures (Beresneva, 2016) Analysis of negative curvature regions in the model's log probability (Mitchell et al., 2023; Bao et al., 2023) However, these statistics-based methods often assume white-box access to model prediction distributions, limiting their applicability to models behind APIs, such as ChatGPT.

Neural Network-Based Detectors An alternative paradigm involves training neural-based detectors (Bakhtin et al., 2019; Fagni et al., 2020; Uchendu et al., 2020; Feng et al., 2021; Tolstykh et al., 2024). These approaches leverage deep learning to identify subtle patterns distinguishing machine-generated text from human-written content. The MAGE project (Li et al., 2024) aggregates a large corpus of machine-generated text but lacks comprehensive coverage of closed-source model outputs.

In conclusion, while the field of MGT detection encompasses a wide range of approaches, from binary and multi-class classification to linguistic pattern analysis and neural network-based detection, significant gaps remain. Most studies have not adequately addressed black-box model detection methods or considered the impact of generation parameters on the detectability of machine-generated text. Our research aims to address these crucial areas, focusing on developing robust detection methods for black-box models and examining how various generation parameters influence the detectability of MGTs.

3 Random sampling Machine-generated Text Detection Problem

Given a text T, for an machine-generated text detection model, the task is to predict and determine whether the text is written by a human or generated by machine. For a traceable machine-generated text detection model, a candidate model list $M = \{m_1, m_2, \ldots, m_n\}$ is provided, and the model can predict whether the text was written by a human or generated by one of the models m_i. With multi-parameter, a given text T may be generated by machine m under the parameter state $Parm$ if it is generated by a machine. In the scope of our discussion, $Parm$ contains Temperature $temp$ and Top-p sampling $topp$. At this point T is denoted as $T_{m_{pram}}$. T is denoted as $T_{m_{greedy}}$ if it is generated from m non-sampled states (greedy).

Big language models are still based on the rule of predicting the next token. The words with the highest probability can be directly selected without sampling, which may lead to overly monotonous and repetitive generated text. Parameters such as top-p, top-k, temperature, etc. are often used to control the randomness of the generated text:

- **Top-p sampling** At each step, only the smallest set of words whose cumulative probability exceeds a certain threshold p is randomly sampled, regardless of other low-probability words.
- **Top-k sampling** At each step, only the k words with the highest probability are randomly sampled, regardless of other low-probability words. Since ChatGPT's api does not support sampling with top-k, we do not discuss this sampling method.
- **Temperature sampling** At each step, the model transforms logits through a distribution to get a new probability distribution, which is then randomly selected. the larger the temperature, the more uniform the new probability distribution.

In order to improve the diversity, generalization and creativity of the large language model, the actual use of the model will often use one or more sampling methods, which has caused some trouble to the machine text detection. In reality, in some occasions of pursuing text accuracy and rigor, the sampling will be set more stable and conservative, Temperature will be set 0 to 0.7, and top-p will be set 0.3 to 0.7, while in some occasions of pursuing novelty and creativity, the sampling will be more random, Temperature will be set 0.7 to 1, and top-p will be set 0.7 to 0.9.

We traversed a total of 16 parameter combination types for Temperature and Top-p of [0.3, 0.5, 0.7, 0.9] respectively, which basically cover all the utilization scenarios in practical situations. And after generating these texts, we computed their similarity with the greedily-generated texts from multiple models. Large language models tend to include prompts in their outputs. To exclude the influence of prompts when calculating similarities, we removed prompt-similar content to achieve more accurate measurements.

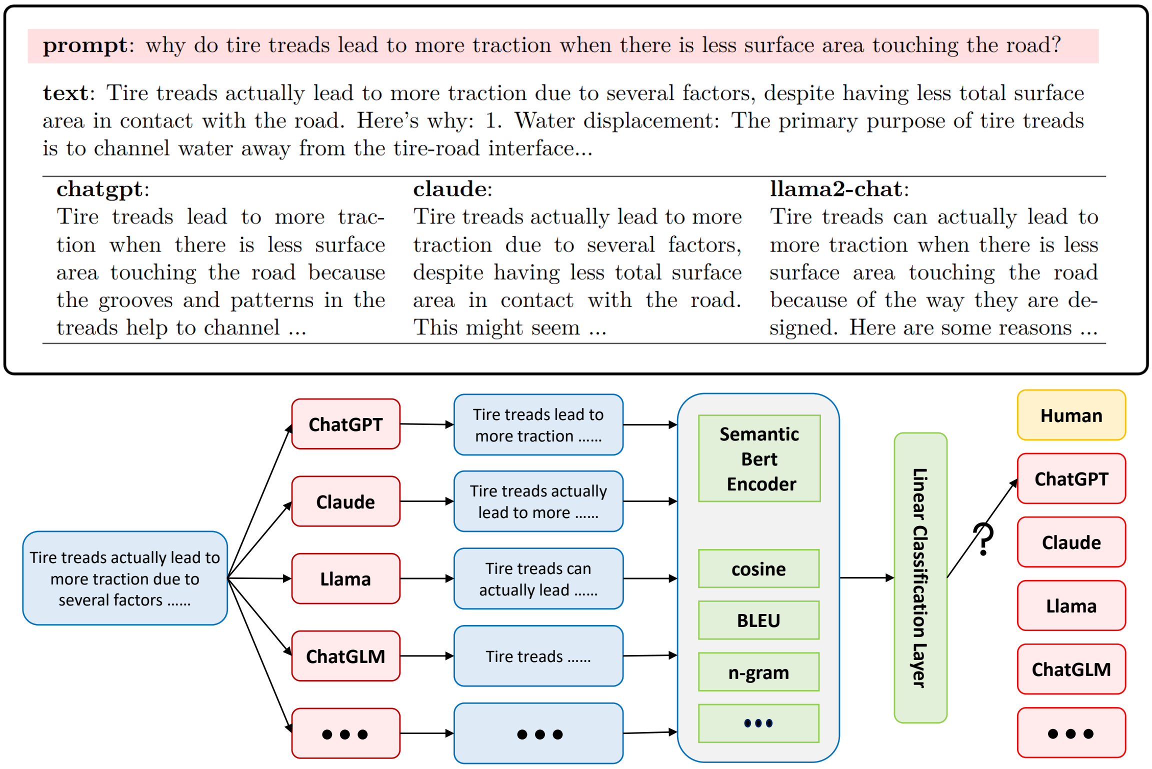
	Temperature			
Top-p	**0.3**	**0.5**	**0.7**	**0.9**
0.3	5.15	5.31	5.01	5.28
0.5	5.35	4.88	4.53	4.29
0.7	4.45	3.89	4.32	3.97
0.9	4.89	3.72	3.24	3.31

Table 1: Relative 1-gram Jaccard Similarity Between ChatGPT's Generated Texts and Greedy Baseline Across Different Temperature and Top-p Settings

	Temperature			
Top-p	**0.3**	**0.5**	**0.7**	**0.9**
0.3	0.60	0.60	0.59	0.60
0.5	0.59	0.61	0.61	0.61
0.7	0.62	0.63	0.63	0.63
0.9	0.61	0.59	0.59	0.63

Table 2: Normalized 1-gram Jaccard Similarity Between Human Text and Model Outputs Across Different Temperature and Top-p Configurations.

Take the 1-gram Jaccard similarity for example. Given two texts A and B, the Jaccard similarity is defined as the size of the intersection divided by



prompt: why do tire treads lead to more traction when there is less surface area touching the road?

text: Tire treads actually lead to more traction due to several factors, despite having less total surface area in contact with the road. Here's why: 1. Water displacement: The primary purpose of tire treads is to channel water away from the tire-road interface...

chatgpt:	**claude**:	**llama2-chat**:
Tire treads lead to more traction when there is less surface area touching the road because the grooves and patterns in the treads help to channel ...	Tire treads actually lead to more traction due to several factors, despite having less total surface area in contact with the road. This might seem ...	Tire treads can actually lead to more traction when there is less surface area touching the road because of the way they are designed. Here are some reasons ...

Tire treads actually lead to more traction due to several factors ......
ChatGPT
Claude
Llama
ChatGLM
•••
Tire treads lead to more traction ......
Tire treads actually lead to more ......
Tire treads can actually lead ......
Tire treads ......
•••
Semantic Bert Encoder
cosine
BLEU
n-gram
•••
Linear Classification Layer
?
Human
ChatGPT
Claude
Llama
ChatGLM
•••

Figure 1: Rewritten Text Demonstration and the MAGRET Framework. MAGRET obtains greedy rewritten text from multiple models by request and predicts whether the text originates from a human or from either model after a semantic encoder and multiple similarity calculations.

the size of the union of the sets representing the words in each text.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

In order to make the word usage preferences of different models more obvious, the vocabulary in prompt P can be removed when calculating the Jaccard similarity. The Jaccard similarity we use is computed like this $J'(A,B) = J(A-P, B-P)$.

Table 1 shows the 1-gram Jaccard similarity between the ChatGPT's parameterized generation and its greedy output to the average similarity with other models' greedy output. Table 2 shows the Cohen's d of human-model similarity to the average cross-model similarity. As can be seen from the Tables, no matter what the parameters are, the model and its own greedily-generated text remain more similar (reflected in table values greater than 1), while the human text consistently remains less similar to the machine text (reflected in table values less than 1), and under more stochastic parameter settings with greater temperate and top-p, the generation of the LLMs among the texts are more similar, but still maintain certain word preferences. However, the similarity between human and machine texts varies minimally with the parameters of machine generation. This predicts that with more random parameter settings, the difficulty of distinguishing between human and machine hardly increases, but the difficulty of tracing which machine the text came from rises.

4 MAGRET: Machine-generated Text Detection with Rewritten Texts

As illustrated in Figure 1, MAGRET employs a multi-stage approach for source prediction of textual content. The methodology initiates with prompt reconstruction, followed by input into multiple generative models to obtain rewritten variations of the original text. Subsequently, a comprehensive analytical framework is applied, involving semantic analysis and statistical characterization of both the original and generated texts. These processed text features are then fed into a fully connected neural network layer, which produces the final classification outcome.

MAGRET contains both semantic and statistical analysis. The semantic analysis consists mainly of a BERT encoder fine-tuned by contrast learning. While the statistical analysis consists of several text rewriting models and a similarity calculation tool.

4.1 Detection with Semantic Analysis

Inspired by the paper (Zheng et al., 2024), the quality of the generated text of a large language model can be evaluated with a stable score from GPT4. From the opposite direction, the semantic quality of the text can be used to predict whether the text is generated by a human or a machine. With multi-parameter, texts generated by the same model can be semantically aligned, we use a comparative learning approach to fine-tune the BERT model. The encoding of the sampled generated text $T_{m_{Para}}$ and the greedily generated text $T_{m_{greedy}}$ of the same model are used as positive sample pairs, while the sampled generated text $T_{m_{Para}}$ and the greedily generated text of a different model $T_{M_{greedy}}, M \neq m$ are used as negative sample pairs. In addition, human-written texts and all greed-generated texts are also used as negative sample pairs.

BERT encoder trained and fine-tuned by comparative learning already has the ability to predict the source of the text without the need for other data in the inference process. This is the baseline that predict multi-parameter generated text at the lowest cost. In the following, we will further introduce how MAGRET can use rewritten text to greatly improve its ability to predict the source of text, if it is available.

4.2 Detection with Rewritten texts

MAGRET request greedy rewritten texts $T' = \{T'_{m_{1greed}}, T'_{m_{2greed}}, \dots, T'_{m_{n greed}}\}$ from the models in M with the rebuild prompt P. By analyzing the semantic and statistical relationships between the text T and the rewritten texts, the neural network predicts which model generated the text. It is worth mentioning that the addition of rewritten text to the input data also enhances the classification ability of the binary MGT.

Reconstructing the prompt and requesting it from the model is an important detection pre-step for this model. We assume several scenarios here and use them to design several methods for obtaining the rewritten text.

- In a Q&A-like environment, subjects may use a question or a question variant as a prompt to obtain a machine-generated answer, in which case the detector can use the question as a prompt to obtain the rewritten text directly.

- In a scenario similar to academic paper writing, the detector can acquire large pieces of text to be detected. If the author uses machine-generated text, the detector can not restore the prompt used by the author at all, at this time, intercept part of the text, and use continuation as the prompt instruction, which can obtain the rewritten text compared with the original text.

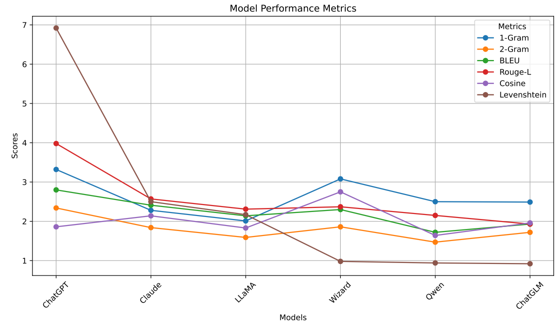


Figure 2: Cohen's d Scores including Jaccard (1-gram and 2-gram), Rouge-L, BLEU, Cosine, and Levenshtein, with averages.

- In the scenario where the topic is clear but the length of the text is short, the detector only has a short text to be detected and is not sure about the form of prompt used by the author, the detector can first input the text to be detected into the large language model to let it summarize and then submit the summarized content to the model to generate the topic text, which can be regarded as a rewritten text.

- Another completely general approach is to directly input the text to be detected into the model and give instructions for it to embellish and rewrite it, treating the returned text as rewritten text.

To mitigate the influence of randomness, the detection model requests greedily generated text from each large language model (LLM) during its reasoning process. By focusing on greedily generated text, we can avoid the complications associated with encoding alignment and similarity matching of randomly generated text. Consequently, all subsequent references to 'rewritten text' in this paper refer specifically to greedily generated text unless otherwise indicated.

We counted the 1-gram, 2-gram, cosine, etc. similarities between the greedy rewrite text and the randomly generated text for a variety of parameters, and calculated the Cohen's d effect, which is plotted in a table as Figure 2. The larger the Cohen's d is, the more the similarity is used to differentiate between the randomly generated text

Dataset	Model	Human	ChatGPT	Claude	LLaMA	Wizard	Qwen	GLM	Macro Avg
SQuAD	SeqXGPT	0.346	0.829	0.786	**0.990**	**0.998**	0.857	0.730	0.791
	MAGRET	**0.822**	**0.956**	**0.980**	0.941	0.970	**0.885**	**0.888**	**0.920**
	w/o Sem	0.796	0.944	0.976	0.921	0.954	0.864	0.876	0.905
	w/o ReT	0.787	0.886	0.710	0.829	0.941	0.578	0.550	0.754
ELI5	SeqXGPT	0.597	0.865	0.933	**0.999**	**0.998**	**0.993**	**0.988**	0.911
	MAGRET	**0.949**	**0.987**	**0.993**	0.971	0.945	0.915	0.921	**0.954**
	w/o Sem	0.933	0.982	0.993	0.952	0.937	0.892	0.900	0.941
	w/o ReT	0.872	0.902	0.904	0.905	0.568	0.611	0.567	0.761
Yelp	SeqXGPT	0.678	0.949	0.935	**0.997**	**0.994**	**0.974**	**0.981**	0.930
	MAGRET	**0.952**	**0.992**	**0.978**	0.978	0.969	0.954	0.948	**0.967**
	w/o Sem	0.940	0.989	0.971	0.964	0.920	0.896	0.872	0.936
	w/o ReT	0.933	0.974	0.916	0.954	0.914	0.871	0.928	0.927

Table 3: Performance of Multiclass Machine-generated Texts Detection. Performance is evaluated with the F1 score. The best results are bolded. Sem: Semantic Encoder. ReT: Rewritten Texts.

attributes. As can be seen from the table, Cohen's d is particularly prominent for 1-Gram, Rouge-L, and Levenshtein. After we obtain the rewritten text, calculate the similarity value between the text to be judged and the rewritten text, and connect the full connectivity layer, the model can predict the human text by statistical scores at the same time.

Alternatively, in addition to statistical methods, the BERT encoder after contrast learning can also be used for rewritten text similarity calculation. Calculating the cosine similarity between the encoded texts can further widen the gap between the texts of different models and narrow the gap between the texts with different parameters.

5 Experiments

Dataset Construction To ensure a fair experimental comparison, we adopted the same sampling and prompt design as in the article (Lu et al., 2023), encompassing three tasks: academic essay writing (Writing), open-ended question answering (QA), and fake review generation (Review). We directly utilized their published dataset, which comprised 20 training samples, 30 validation samples, and 200 test samples across these tasks. These tasks correspond to the SQuAD (Rajpurkar et al., 2016), Eli5 (Fan et al., 2019), and Yelp (Zhang et al., 2015) datasets, respectively, utilizing a data partitioning approach where training data is significantly smaller than test data. This prevents F1 scores of 1 in training results, which complicate comparative analysis. In practice, as an AI-generated content detection model, the amount of text to be detected far exceeds the training text, making these results more relevant.

For the Writing task, we truncated the first 30 characters and filled in the rest using an AI model. In the QA task, we input question Q into the large model to obtain AI-generated samples. For the Review task, human text is first summarized by the large model, and then the summary is rewritten.

We selected ChatGPT-3.5, Claude, LLaMA (7B), ChatGLM (9B), Wizard (7B), and Qwen (7B) as the LLMs for text generation. ChatGPT-3.5 and Claude were accessed via API, while LLaMA (7B), ChatGLM (9B), Wizard (7B), and Qwen (7B) were run locally. We generated text under various random sampling parameters, with greedy text generation settings of do_sample=False, temperature=1e-10, and top_p=1. The parameters for random sampling included temperatures and top-p values ranging from 0.3 to 0.9. We refer to the collected datasets as MAGRET-Bench.

Implementation Details In MAGRET-Bench, ChatGPT-3.5 and Claude are closed-source models, while LLaMA (7B), ChatGLM (9B), Wizard (7B), and Qwen (7B) are open-source models, allowing for weight retrieval in the white-box setting. MAGRET can access all rewritten texts. Each open-source model operates on a separate GPU, with the main program creating a Flask service for local API calls, setting the maximum token generation limit to 1000. During training, MAGRET-Bench includes semantic prediction and statistical prediction modules, each trained for 100 epochs, selecting the best weights from the evaluation dataset, followed by an additional 50 epochs to integrate the prediction results. For the MAGRET model, we combined the machine outputs from the multi-classification task into binary classification results.

We selected SeqXGPT (Wang et al., 2023) as our baseline, as it represents the state-of-the-art in AI-generated text detection. While SeqXGPT demonstrates strong detection capabilities for white-box models, it has also achieved remarkable performance on closed-source models using out-of-distribution techniques. SeqXGPT also converged after 100 epochs of training.

We designed two ablation experiments: one assessing MAGRET's prediction capability without accessing rewritten texts, termed w/o ReT, and the other evaluating MAGRET's performance without the semantic encoder, referred to as w/o Sem.

In the evaluation process, we used the Macro-F1 Score as our metric, effectively combining Precision and Recall, allowing us to consider the overall performance.

5.1 Results in Multiclass Machine-generated Texts Detection

We evaluated the text traceability performance of multiple models in a multi-random sampling and multi-model environment. This evaluation assessed not only the models' ability to detect machine-generated text but also to identify the specific large model from which the text originated. Results are presented in Table 3. MAGRET demonstrated the best performance in detecting closed-source models. In contrast, SeqXGPT achieved high scores in open-source model detection but exhibited instability when predicting the performance of the closed-source models ChatGPT and Claude through out-of-distribution assessments. SeqXGPT also struggled with distinguishing human text from closed-source model text, particularly in the more open-ended Writing (SQuAD) task. Additionally, while the result without Semantic Encoder generally produced good predictions, its performance was highly unstable. This instability in performance appears to be largely independent of the model used but significantly influenced by specific datasets. The performance without rewritten texts on the Yelp dataset was notably better than on the other two datasets, likely due to the clear themes and rich semantic information present in the Review task. In contrast, methods based on rewritten text consistently maintained strong performance.

5.2 Results in Binary Machine-generated Texts Detection

In some tasks, there is a greater focus on whether the text is machine-generated or human-written. We conducted tests under multi-model and multi-random sampling conditions, with results presented in Table 4. SeqXGPT showed improved binary classification performance compared to multi-class classification, likely due to the integration of texts from the closed-source models ChatGPT and Claude. MAGRET's performance remained largely consistent with that in multi-class classification. Notably, text detection in the Writing task posed challenges for all models, likely because many continuation models default to copying or refining previous text in continuation tasks. This results in machine-generated text exhibiting some human-like features, thereby impacting detection performance.

5.3 Results in Random Sampling Generated Texts Detection

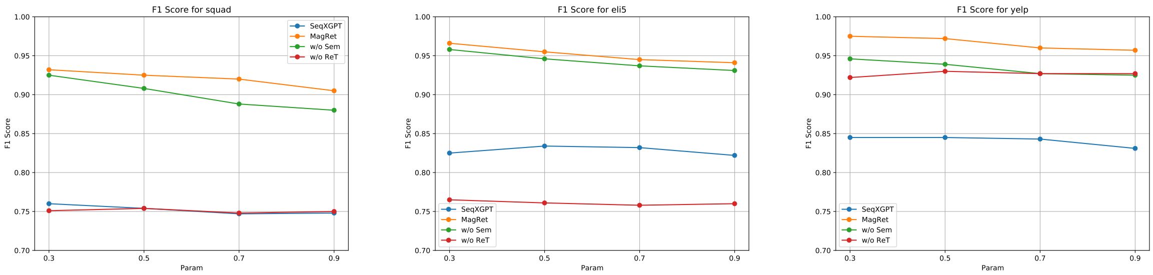
We conducted a statistical analysis of F1 scores for texts generated using various random sampling parameters in a multi-class classification scenario. This experiment aimed to investigate the impact of different random parameters on the model's detection capabilities. As illustrated in Figure 3, the results without semantic Semantic Encoder (w/o Sem) indicate that texts generated through increasingly random sampling methods become statistically more challenging to differentiate. This phenomenon may be attributed to the uniform distribution of word usage probabilities, leading to diminished statistical differences in the generated texts. Conversely, the results without rewritten texts (w/o Ret) reveal that random sampling has minimal effect on the semantic alignment approach. The SeqXGPT method, which leverages information from open-source models, shows a slight decline in performance with increased randomness, although it remains relatively unaffected overall.

5.4 Results in Out-of-distribution Detection

Accessing rewritten texts requires requesting APIs from closed-source models, which may impose financial burdens on users. Consequently, we evaluated our model's ability to predict out-of-distribution instances when limited to rewritten texts from open-source models. This experiment establishes a baseline performance for MAGRET and offers users a cost-effective prediction method. As shown in Table 5, even without direct access to rewritten texts from closed-source models, the statistical relationship between rewritten texts from open-source models and those from closed-source models (w/o Sem) still provides stable predictive capabilities. Our experiments demonstrate that when predicting out-of-distribution instances for closed-source models, using rewritten texts from open-source models outperforms generating probabilities. This improved performance may result

Dataset	Model	Human	Machine	Macro Avg
SQuAD	SeqXGPT	0.489	0.976	0.732
	MAGRET	0.822	0.937	0.880
	w/o Sem	0.796	0.922	0.859
	w/o ReT	0.787	0.749	0.768
ELI5	SeqXGPT	0.694	0.996	0.845
	MAGRET	0.949	0.955	0.952
	w/o Sem	0.933	0.943	0.938
	w/o ReT	0.872	0.743	0.808
Yelp	SeqXGPT	0.678	0.989	0.833
	MAGRET	0.952	0.970	0.961
	w/o Sem	0.940	0.935	0.938
	w/o ReT	0.933	0.926	0.930

Table 4: Results of Binary Machine-generated Texts Detection. Performance is evaluated with the F1 score. Sem:Semantic Encoder. ReT:Rewritten Texts.



(a) Squad

(b) Eli5

(c) Yelp

Figure 3: Results of different random sampling texts detection in Squad, Eli5 and Yelp Datasets with F1 scores. The horizontal coordinate is the value of the random sampling parameter (Temperature, Top-p)

Dataset	Model	Human	ChatGPT	Claude	Macro Avg
SQuAD	SeqXGPT	0.346	0.829	0.786	0.654
	MAGRET	**0.862**	**0.893**	**0.803**	**0.853**
ELI5	SeqXGPT	0.597	0.865	0.933	0.798
	MAGRET	**0.887**	**0.895**	**0.945**	**0.909**
Yelp	SeqXGPT	0.678	0.949	0.935	0.854
	MAGRET	**0.895**	**0.966**	**0.957**	**0.939**

Table 5: Results in Out-of-distribution Detection. The best results are bolded. Sem:Semantic Encoder. ReT:Rewritten Texts.

from the coherent nature of rewritten texts, which contain more semantic and statistical information than generated probabilities.

6 Conclusion

In this paper, we demonstrate that text can be classified as machine-generated based solely on multiple features of similar natural languages, allowing for the identification of the generating model. We introduce MAGRET, a novel method for detecting machine-generated text using complete greedy rewritten texts. Variations in generated texts under different random generation parameters can impact detection performance. We discuss in detail the effects of two parameters, Temperature and Top-p, on generated texts, and our experiments confirm that predictions based on rewritten texts maintain substantial detection performance even for highly randomized outputs. Experiments conducted on texts from humans and seven LLMs showcase the superiority of MAGRET in binary, multiclass, and out-of-distribution (OOD) scenarios.

Limitations

Despite MAGRET exhibits excellent performance in close-source machine-generated text detection, it still present certain limitations:

- Our method relies on the availability of greedy rewritten texts from models. Accessing APIs from closed-source models requires a constant internet connection and ongoing payments, which limits the versatility of MAGRET. However, we demonstrate that MAGRET retains a certain level of detection performance even with greedy rewritten texts obtained solely from open-source models.

- The prerequisite for acquiring rewritten texts is the ability to reconstruct prompts, which restricts the application of our model to texts of sufficient length. A minimum text length is necessary for effective segmentation, rewriting, and semantic analysis; our tests indicate that texts exceeding 500 words can fully leverage MAGRET's capabilities.

- We focused only on parameter configurations for commonly used machine-generated texts, without exploring additional random sampling methods and parameter settings. We plan to conduct more comprehensive experiments in future work to examine the impact of random sampling on machine-generated text detection.

Acknowledgement

This work is supported by National Natural Science Foundation of China under Grants No.62472092, No.62172089. Natural Science Foundation of Jiangsu province under Grants No.BK20241751. Jiangsu Provincial Key Laboratory of Computer Networking Technology. Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No.BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No.93K-9, Nanjing Purple Mountain Laboratories. Startup Research Fund of Southeast University under Grants No.RF1028623097. We thank the Big Data Computing Center of Southeast University for providing the facility support on the numerical calculations.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

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.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *ArXiv*, abs/2310.05130.

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *International Conference on Applications of Natural Language to Data Bases*.

Yu-Chu Chang, Xu Wang, Jindong Wang, Yuanyi Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Weirong Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qian Yang, and Xingxu Xie. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15:1 – 45.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*.

Tiziano Fagni, F. Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2020. Tweepfake: About detecting deepfake tweets. *PLoS ONE*, 16.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *ArXiv*, abs/1907.09190.

Chen Feng, Georgios Tzimiropoulos, and Ioannis Patras. 2021. Ssr: An efficient and robust framework for learning with unknown label noise. *arXiv preprint arXiv:2111.11288*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

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. *ArXiv*, abs/2301.07597.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *ArXiv*, abs/2307.03838.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Pan*.

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. 2023. Origin tracing and detecting of llms. *arXiv preprint arXiv:2304.14072*.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Yu Lan, and Chao Shen. 2022. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *ArXiv*, abs/2212.10341.

Ning Lu, Shengcai Liu, Rui He, Qi Wang, Yew-Soon Ong, and Ke Tang. 2023. Large language models can be guided to evade ai-generated text detection. *arXiv preprint arXiv:2305.10847*.

Niloofar Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2024. Smaller language models are better zero-shot machine-generated text detectors. In *Conference of the European Chapter of the Association for Computational Linguistics*.

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*.

Shaoor Munir, Brishna Batool, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar. 2021. Through the looking glass: Learning to attribute synthetic text generated by language models. In *Conference of the European Chapter of the Association for Computational Linguistics*.

OpenAI. 2023. Chatgpt release notes. Accessed: 2024-09-10.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

Yuhui Shi, Qiang Sheng, Juan Cao, Hao Mi, Beizhe Hu, and Danding Wang. 2024. Ten words only still help: Improving black-box ai-generated text detection via proxy-guided efficient re-sampling. *ArXiv*, abs/2402.09199.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. 2021. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*.

Irina Tolstykh, Aleksandra Tsybina, Sergey Yakubson, Aleksandr Gordeev, Vladimir Dokholyan, and Maksim Kuprashevich. 2024. Gigacheck: Detecting llm-generated content. *arXiv preprint arXiv:2410.23728*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Conference on Empirical Methods in Natural Language Processing*.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296*.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector. *ArXiv*, abs/2310.06202.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. Seqxgpt: Sentence-level ai-generated text detection. *arXiv preprint arXiv:2310.08903*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. 2024. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. *arXiv preprint arXiv:2402.11175*.

Feng Xiong, Thanet Markchom, Ziwei Zheng, Subin Jung, Varun Ojha, and Huizhi Liang. 2024. Fine-tuning large language models for multigenerator, multidomain, and multilingual machine-generated text detection. *ArXiv*, abs/2401.12326.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Neural Information Processing Systems*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. *arXiv preprint arXiv:2010.07475*.