SearchLLM: Detecting LLM Paraphrased Text by Measuring the Similarity with Regeneration of the Candidate Source via Search Engine

Hoang-Quoc Nguyen-Son[†], Minh-Son Dao[†], and Koji Zettsu [†‡]
[†]National Institute of Information and Communications Technology, Japan
{quoc-nguyen,dao,zettsu}@nict.go.jp
[‡] Nagoya University, Japan
zettsu@i.nagoya-u.ac.jp

Abstract

With the advent of large language models (LLMs), it has become common practice for users to draft text and utilize LLMs to enhance its quality through paraphrasing. However, this process can sometimes result in the loss or distortion of the original intended meaning. Due to the human-like quality of LLM-generated text, traditional detection methods often fail, particularly when text is paraphrased to closely mimic original content. In response to these challenges, we propose a novel approach named SearchLLM, designed to identify LLM-paraphrased text by leveraging search engine capabilities to locate potential original text sources. By analyzing similarities between the input and regenerated versions of candidate sources, SearchLLM effectively distinguishes LLM-paraphrased content. SearchLLM is designed as a proxy layer, allowing seamless integration with existing detectors to enhance their performance. Experimental results across various LLMs demonstrate that SearchLLM consistently enhances the accuracy of recent detectors in detecting LLM-paraphrased text that closely mimics original content. Furthermore, SearchLLM also helps the detectors prevent paraphrasing attacks.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in supporting a wide range of linguistics-related tasks. Despite their power and versatility, the increasing reliance on LLMs also raises concerns regarding potential misuse, particularly when users employ these models to paraphrase texts without adequately reviewing the generated outputs. This practice can result in the final text failing to accurately convey the original intent, potentially leading to misunderstandings among recipients. As a result, there is an urgent need for robust detection mechanisms to identify and differentiate LLM-paraphrased content from genuine human writing.

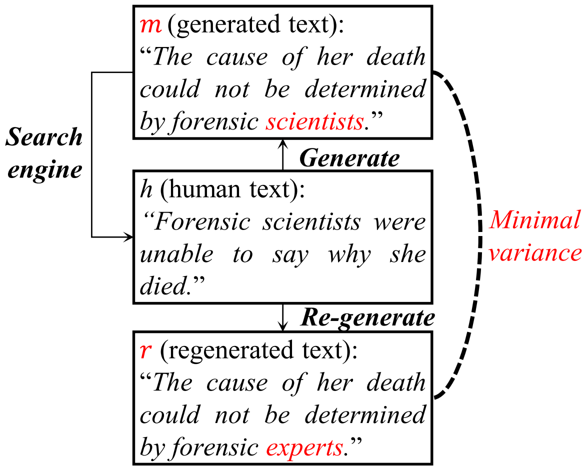

Figure 1: Illustration of variance in LLM-generated texts derived from a human text h. The comparison demonstrates minimal variance between the generated text m and the regenerated text r.

Exisiting LLM detectors can be categorized into three primary approaches. The watermarking approach involves embedding specific characteristics within the generated text, allowing for differentiation between human-authored and LLM-generated content (Kirchenbauer et al., 2023; Chang et al., 2024; Gloaguen et al., 2025). However, this method faces limitations with closed LLM models where such control is not feasible. The second approach leverages supervised learning, utilizing extensive datasets of both human and LLM-generated texts to train models that can distinguish between the two (Hu et al., 2023; Li et al., 2024; Xu et al., 2024). While effective, this method is sensitive to out-of-distribution scenarios, which can compromise its reliability. The third approach employs zero-shot techniques, enabling detection without prior training by exploiting inherent differences in text structure and style (Mitchell et al., 2023; Hans et al., 2024; Park et al., 2025).

Despite these varied strategies, detecting LLM-generated text that closely mimics human writing

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1755–1772

remains a formidable challenge. Consequently, we propose focusing on zero-shot methods to enhance current detection systems, aiming to address the nuanced problem of identifying text that seamlessly integrates into human-like narratives. This proposal builds on existing frameworks and seeks to fortify the detection capabilities against increasingly sophisticated LLM outputs.

Motivation: In our investigation, we observe a notable consistency in the outputs generated by a specific LLM across multiple trials. For instance, a human-authored text, denoted as h, was randomly selected from the XSum dataset (Narayan et al., 2018), and GPT-4o-mini was utilized with default temperature settings to produce a paraphrased version, m, as shown in Figure 1 with the prompt adapt from Zhu et al. (2023): "*Paraphrase the following text: <human text>*." Subsequently, employing the same model, a re-generated text, r, was created from h. The striking similarity between m and r suggests that if the original human text h can be retrieved, via search engine capabilities, it becomes feasible to detect the paraphrased output m.

We compare the similarity between 1,000 human-written samples and their corresponding sources retrieved from the internet, as well as 1,000 LLM-paraphrased samples and their sources, as shown in Figure 2. The results indicate that human-written samples generally exhibit higher similarity to their sources than the LLM-paraphrased samples. However, a few human-written samples show low similarity, likely due to noise in the internet sources, such as updates since their initial creation or issues with inconsistent parsing by crawling tools. To further investigate these cases, we regenerate both the human and LLM samples using GPT-4o-mini and analyze the change in similarity between the regenerated and human/LLM samples, as shown in Figure 3. Since GPT-4o-mini tends to regenerate LLM-paraphrased text that is more similar to the internet source, the similarity shift in these cases is generally positive. In contrast, when GPT-4o-mini regenerates human-written text that diverges from the internet source, the similarity shift tends to be negative.

Contribution: We introduced a method called SearchLLM to identify text paraphrased by LLMs. The method operates by initially utilizing search engines to identify candidate original texts corresponding to the input text. Subsequently, SearchLLM produces re-generated texts from these candidate originals. By comparing the input text with the re-generated versions, SearchLLM can effectively discern whether the text is authored by a human or paraphrased by an LLM. Our contribution can be summarized as follows:

- We propose a method called SearchLLM, which can detect text paraphrased by LLMs.
- We design SearchLLM to function as a proxy, allowing it to enhance the performance of any existing detectors[1].
- We conducted experiments with various datasets, which demonstrate that SearchLLM effectively improves the performance of existing detectors. Additionally, SearchLLM efficiently helps these detectors identify text that closely mimics original content and are consistent with paraphrasing attacks.

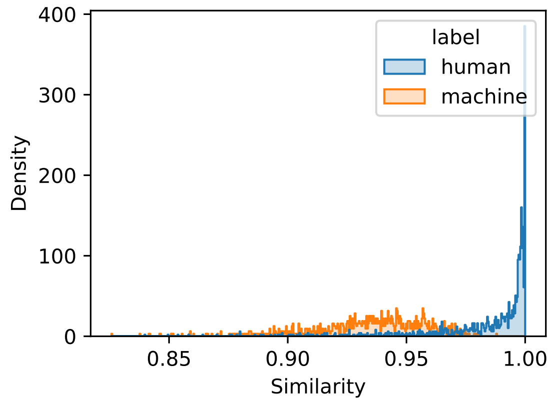

Figure 2: Comparison of similarity between samples produced by humans and LLMs with internet sources.

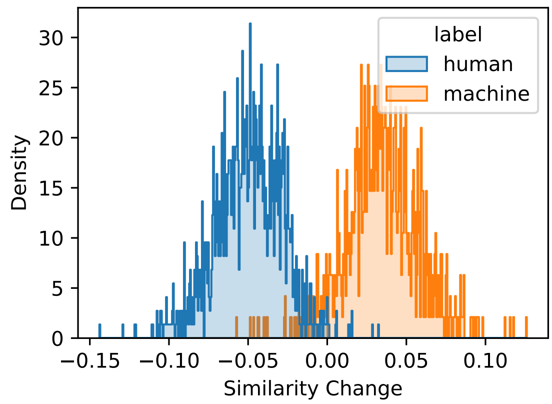

Figure 3: Change in similarity between regenerated samples and human or LLM-generated samples.

[1] Source code is available at https://github.com/quocnsh/SearchLLM

2 Related Work

The LLM text detection methods can be categorized into three main approaches: watermarking, supervised-based, and zero-shot.

Watermarking: An LLM is guided to produce outputs that contain specific, embedded characteristics (Gloaguen et al., 2025; Lu et al., 2024; Huo et al., 2024; Chang et al., 2024). These characteristics act as subtle signals which can help distinguish LLM-generated text from human-authored content. For instance, Kirchenbauer et al. (2023) demonstrated a method where the LLM is instructed to preferentially use terms from a "green" word list while avoiding those from a "red" word list, thereby embedding a detectable linguistic signature within the output. Such watermarking not only enables highly effective detection of LLM-generated text, but also provides verifiable evidence to support the identification process.

Widespread adoption of techniques to safeguard media authenticity is evident, such as the Coalition for Content Provenance and Authenticity (C2PA) initiative used to watermark AI-generated images like those from GPT Image. However, watermarking text is more challenging due to the context-dependent nature of language and requires direct control over the LLM output, limiting it to closed models. Additionally, watermarking carries risks like piggyback spoofing, where attackers embed watermark patterns into human-written text, undermining detection reliability.

Supervised-Based: The prevailing methods entail collecting large corpora and training a classifier on the final layer of transformer-based models. Moreover, recent findings have demonstrated that leveraging representations from intermediate layers can yield additional benefits for distinguishing subtle textual characteristics (Yu et al., 2024a). Complementary to this, Verma et al. (2024) expand the feature set by integrating n-gram statistics, which enhances the model's expressiveness and robustness. Furthermore, Tian et al. (2024) introduce a multiscale positive-unlabeled detection framework tailored for separately processing short texts, addressing length-based variability in classification performance.

To further improve classification accuracy, some techniques involve generating augmented argument samples through paraphrasing (Hu et al., 2023), thereby enriching the diversity of training data. Contract learning has also emerged as a promising strategy, offering a systematic means to explicitly distinguish between outputs generated by LLMs and those produced by humans (Liu et al., 2024; Guo et al., 2024b; Liu et al., 2023). In addition, the adoption of specialized techniques, such as syntactic parsing trees (Park et al., 2025; Li et al., 2025; Kim et al., 2024) and Fourier transformation (Xu et al., 2024), facilitates the extraction of structural and frequency-based features, deepening the model's analytical capacity.

Zero-Shot: Another prominent approach for detecting LLM-generated text is zero-shot learning, which sidesteps the need for dedicated training by directly leveraging statistical or behavioral properties of LLM outputs. A notable method in this category was proposed by Mitchell et al. (2023), who demonstrated that LLM-generated text tends to be optimized for higher probability under the generation model compared to human-written text. Building on this foundational insight, subsequent methods have sought to improve detection by enhancing either speed (Bao et al., 2024) or detection accuracy (Su et al., 2023; Zeng et al., 2024; Xu et al., 2025). Some techniques further adapt this probabilistic hypothesis to develop black-box detectors by relying on an alternative model (Mireshghallah et al., 2024; Ma and Wang, 2024; Shi et al., 2024; Bao et al., 2025; Wu et al., 2023) or two alternative models (Hans et al., 2024). Additionally, Guo et al. (2024a) exploit the LLM's ability to memorize previously generated words to distinguish machine-generated content, whereas other methods (Koike et al., 2024; Bhattacharjee and Liu, 2024) directly task an LLM with self-identification of its own outputs.

Closely related to our hypothesis, recent approaches recognize that LLMs often produce text with distinct generative patterns, prompting the use of the LLM itself to regenerate a text and compare similarity to the input text (Zhu et al., 2023; Nguyen-Son et al., 2024; Mao et al., 2024; Yang et al., 2024; Yu et al., 2024b). The key distinction from our method is that, while prior work typically employs the input text for regeneration, we draw upon candidate source materials retrieved from the internet as the basis for text regeneration. Nonetheless, both supervised and zero-shot detection techniques are notably sensitive to out-of-distribution texts and are vulnerable to attacks, such as paraphrasing (Krishna et al., 2023), which significantly undermine their robustness in adversarial scenarios.

3 SearchLLM

Overview: Figure 4 illustrates the workflow employed by SearchLLM for detecting text generated by LLMs. Given an input text t, the process starts by utilizing a search engine to retrieve a candidate source text t_c that is related to t and could plausibly serve as its origin. Next, the similarity score σ_c between the original candidate text t_c and the input text t is computed. If σ_c exceeds a predetermined threshold α, the input text t is classified as human-written, under the assumption that significant textual overlap with verifiable sources indicates human authorship. If σ_c does not surpass the threshold, the method proceeds by leveraging the LLM to re-generate a new text t_r based on the candidate t_c. The similarity between t_r and the original input t is then calculated; a sufficiently high similarity score at this stage suggests that t was likely generated by an LLM. In cases where neither condition is conclusive, the detection of t is deferred to an existing detection method. The details of the process are described in Algorithm 1 of Appendix A. The main steps of SearchLLM for detecting LLM-generated text are summarized as follows:

Candidate Extraction: This step assesses whether the text at a given URL u could serve as a potential input source for generating the target text t. We observe that input text may be rewritten from the original source. After rewriting, some sentences may be merged or separated. Therefore, we propose a greedy matching approach between the input text and the retrieved source, as illustrated in Figure 5.

Both t and the text from u are segmented into individual sentences. First, we identify the sentence s_{u_1} from u that has the highest similarity to the first sentence s_{t_1} in the input text, serving as an anchor. Similarity is measured using a fixed pretrained model[2]. Starting from the anchor, we find the optimal matching between the first sentences in u and the first sentence s_{t_1}. We also compare the optimal matching between the first sentences in t and s_{u_1}. The best matching is retained. This process is repeated until all sentences in t are matched. The matching process is demonstrated in Algorithm 2 of Appendix A.

Similarity Measurement: We estimate similarity based on the matching method described above. Specifically, we evaluate sentences with similarity scores greater than the machine threshold β, which is fixed at 0.86[3]. If the number of remaining sentences exceeds the ratio threshold γ at 0.5 and their average similarity is greater than the human threshold α (set at 0.99), we confidently classify t as human-written. Otherwise, we use the prompt "*Paraphrase the following text: <candidate text* t_c>" to generate a regenerated version of the text, t_r, based on matched sentences in u, and calculate a new similarity score σ_r between t and t_r using the filtered sentences. If the increase from σ_c to σ_r exceeds the predefined threshold Δ (where $\Delta = 0.01$), we classify t as LLM-generated. If neither condition is met, the input t is subsequently passed to an existing detection method for further analysis.

4 Evaluation

4.1 General Scenario

We conduct experiments on the RAID dataset (Dugan et al., 2024), which contains human-written and LLM-generated texts. We utilize all available Wikipedia-related texts and employ the Wikipedia search engine for the SearchLLM. The LLM text is generated using 11 LLMs from five providers: GPT, LLaMA, Cohere, MPT, and Mistral. Specifically, we use representative models from each provider, including GPT-4, Llama-2-70B-Chat, Cohere-Chat, MPT-30B-Chat, and Mistral-7B-Chat. This process yields a total of 10,794 samples, which are equally distributed between human-authored content and text generated by these five representative LLMs.

We use GPT-4o-mini within SearchLLM to create regenerated texts by paraphrasing. Since the LLM-generated texts in this dataset are produced using topic-based prompts, their meanings often differ from those of human-written texts. Therefore, any LLM used in SearchLLM to create regenerated texts tends to yield same performance.

We evaluate a wide range of representative methods, including RADAR (Hu et al., 2023), Longformer (Li et al., 2024), DetectGPT (Mitchell et al., 2023), and Binoculars (Hans et al., 2024). Watermarking-based detectors are not applicable to GPT-4 in the RAID dataset. Longformer is a supervised

[2] https://huggingface.co/Qwen/Qwen3-Embedding-0.6B

[3] All hyperparameters are determined via beam search with a step size of 0.01 and are held constant for all experiments in this paper.

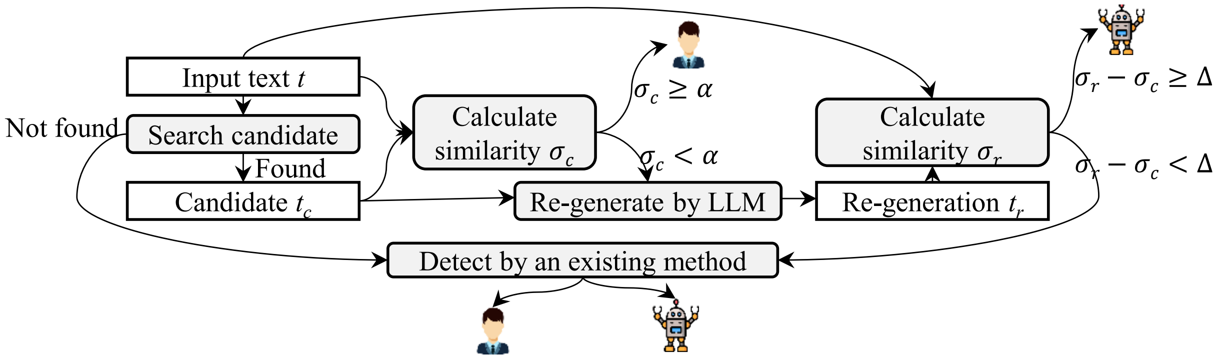

Figure 4: Overview of the SearchLLM schema for determining whether input text t is human- or LLM-generated. The system considers three main cases: (1) SearchLLM compares t with a candidate t_c retrieved by a search engine to identify human text; (2) SearchLLM generates a regeneration t_r from t_c to determine LLM-generated text; (3) if neither case applies, SearchLLM delegates the decision to an existing method.

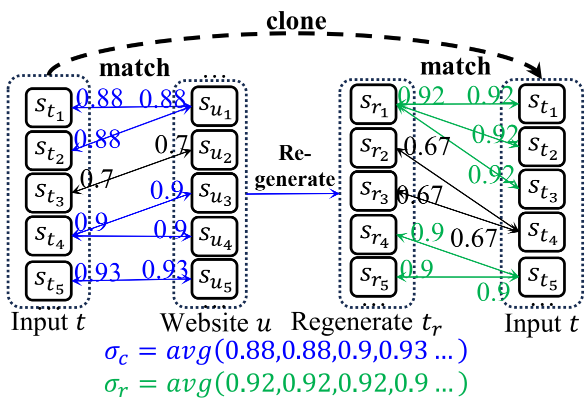

Figure 5: Process of matching between input data and website content or re-generated text.

detector trained on a large volume of text from the MAGE dataset (Li et al., 2024). RADAR uses the Vicuna-7B model, which is trained on an argumentation dataset with paraphrased texts via adversarial learning. DetectGPT estimates changes in probability on the GPT-2-XL model after perturbations generated by the T5-large model. Binoculars measures perplexity using two models: Falcon-7B and Falcon-7B-Instruct. For evaluation, we adopt a ROC AUC metric and ROC AUC at false positive rate's of 1%, standard approaches widely used for assessing the performance of LLM detectors. The F-score is shown in Appendix B.

Table 1 shows that the existing detectors are effective at identifying LLM-generated text across all LLM models. Among the supervised-based detectors, Longformer and RADAR achieve nearly identical performance on each model. In contrast, among the zero-shot-based detectors, DetectGPT performs slightly worse than the supervised detectors, while Binoculars outperforms all existing detectors with a ROC AUC exceeding 0.997. SearchLLM further enhances the performance of all detectors. Specifically, it maintains Binoculars' performance at 0.9999 on LLaMa and MPT models and improves the performance in other cases. In Table 2, SearchLLM achieves competitive ROC AUC at false positive rate's of 1% compared to Binoculars and outperforms the other methods.

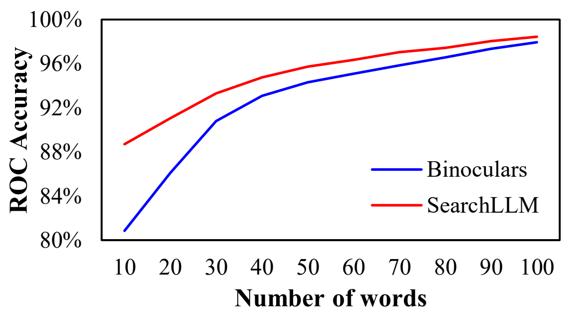

Figure 6: Comparison of Binoculars and SearchLLM in detecting LLM-generated text with limited words.

Short Text: We conduct experiments using short texts, with lengths limited to between 10 and 100 words, as shown in Figure 6. The experiments are performed with GPT-4 text and SearchLLM on the representative method, Binoculars. The results demonstrate that SearchLLM can improve the performance of Binoculars, especially with short texts. In particular, Binoculars combined with SearchLLM achieves a ROC AUC of 89.7%, when the text length is only 10 words.

4.2 Rigorous Scenario

To rigorously assess the ability to detect LLM-generated text that closely resembles human-written content, we design an evaluation scenario in

Method	GPT-4	Llama-2-70B	Cohere	MPT-30B	Mistral-7B
Longformer	0.9932	0.9970	0.9505	0.9916	0.9895
SearchLLM	**0.9979**	**0.9996**	**0.9794**	**0.9975**	**0.9966**
RADAR	0.9906	0.9968	0.9599	0.9962	0.9947
SearchLLM	**0.9949**	**0.9982**	**0.9793**	**0.9979**	**0.9970**
Detect	0.9022	0.9167	0.7629	0.8542	0.8868
SearchLLM	**0.9379**	**0.9546**	**0.8793**	**0.9233**	**0.9399**
Binoculars	0.9987	**0.9999**	0.9977	**0.9999**	0.9998
SearchLLM	**0.9994**	**0.9999**	**0.9989**	**0.9999**	**0.9999**

Table 1: Performance of LLM-generated text detection on all Wikipedia-related samples from the RAID dataset (ROC AUC).

Method	GPT-4	Llama-2-70B	Cohere	MPT-30B	Mistral-7B	Average
Longformer	0.4824	0.4820	0.2505	0.3924	0.4008	0.4016
SearchLLM	**0.8912**	**0.9297**	**0.6102**	**0.7857**	**0.8456**	**0.8124**
RADAR	0.6190	0.7493	0.4241	0.7628	0.7003	0.6511
SearchLLM	**0.7166**	**0.8268**	**0.4863**	**0.8235**	**0.7758**	**0.7258**
DetectGPT	0.0174	0.0340	0.0097	0.0186	0.0292	0.0218
SearchLLM	**0.0639**	**0.0896**	**0.0288**	**0.0502**	**0.0822**	**0.0629**
Binoculars	**0.7736**	0.0561	**0.9233**	0.0561	0.5046	0.4627
SearchLLM	0.7207	**0.6182**	0.8754	**0.6182**	**0.6732**	**0.7011**

Table 2: Performance of LLM-generated text detection on all Wikipedia-related samples from the RAID dataset (ROC AUC at an FPR of 1%).

Method	4o-mini	4.1	DeepSeek	Grok	Phi
Long	0.6552	0.6805	0.4997	0.4996	0.5018
Search	**0.9073**	**0.9173**	**0.8630**	**0.8607**	**0.8145**
RA	0.6543	0.6436	0.6070	0.6038	0.6023
Search	**0.9062**	**0.9107**	**0.8926**	**0.8906**	**0.8473**
Detect	0.6073	0.5572	0.5761	0.5593	0.5751
Search	**0.8876**	**0.8817**	**0.8792**	**0.8779**	**0.8675**
Bino	0.6388	0.6127	0.4327	0.4351	0.4363
Search	**0.9070**	**0.9064**	**0.8481**	**0.8456**	**0.7919**

Table 3: Detection of paraphrased text generated by various large language models.

.

which various large language models (LLMs) are tasked with paraphrasing text originally authored by Wikipedia humans as illustrated in Table 3. Specifically, we utilize a diverse selection of state-of-the-art LLMs, including GPT-4o-mini (4o-mini), GPT-4.1 (4.1), DeepSeek-V3-0324 (DeepSeek), Grok-3-mini (Grok), and Phi-4 (Phi). We use the corresponding LLM in SearchLLM to create the regenerated text. The scenarios involving unknown LLMs and prompts are further detailed later.

The results demonstrate that existing methods experience a significant decline in performance when applied to this rigorous scenario. Specifically, many of the results show a sharp drop in a ROC AUC to around 0.6. In contrast, paired t tests conducted in Appendix C demonstrate that SearchLLM significantly improves the performance of all existing methods across various models, consistently maintaining a ROC AUC above 0.79 in every case.

Breakdown of Performance: We present a breakdown of SearchLLM's performance in conjunction with Binoculars for detecting LLM-generated text by GPT-4o-mini, as shown in Figure 7. When LLM-generated text is paraphrased, it becomes more challenging for the search engine to retrieve than human-written text. Consequently, SearchLLM processes 56.2% of human text, and 43.8% of LLM-generated text. Among the texts that are processed, SearchLLM achieves higher precision, recall, and $F1$ scores overall compared to Binoculars. Similar results for detecting GPT-4.1-generated text are presented in Appendix D.

Ablation Studies: We evaluate the effect of each component of SearchLLM on detecting text generated by GPT-4o-mini, as shown in Table 4. The three main components are: α, for detecting human-written text; β, for detecting LLM-generated text;

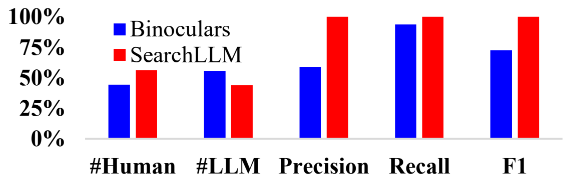

Figure 7: The number of human- and LLM-generated texts processed by SearchLLM and Binoculars, along with their performance in detecting LLM-generated content.

Variant	Search (Long)	Search (RA)	Search (Detect)	Search (Bino)
w/o α	0.7920	0.8068	0.7376	0.7952
w/o β	0.8466	0.8466	0.8021	0.8359
w/o Δ	0.7914	0.7865	0.7561	0.7891
w/o Search	0.6552	0.6543	0.6073	0.6388
Full model	**0.9073**	**0.9062**	**0.8876**	**0.9070**

Table 4: Ablation studies examining the impact of different hyperparameter settings on SearchLLM.

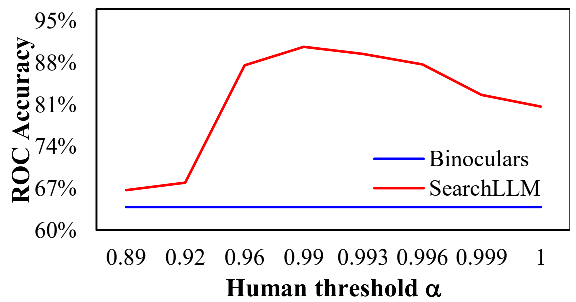

Figure 8: The impact of varying the human threshold parameter α on the performance of SearchLLM.

and Δ, for rechecking the LLM-generated text. Since SearchLLM processes more human-written text than machine-generated text, α is more important than β. Δ also demonstrates the importance of rechecking LLM-generated text. Overall, all variants of SearchLLM still outperform the corresponding existing approaches that do not use SearchLLM.

Change of Parameters: We analyze the impact of varying the human threshold α while keeping all other parameters constant as shown in Figure 8. Other parameters are detailed in Appendix E. When α is set below 0.92, SearchLLM achieves performance comparable to Binoculars. As α increases beyond 0.96, SearchLLM attains more stable performance.

Unknown Models and Prompts: We conduct experiments to detect text generated by large language models (LLMs) when the specific LLM and/or prompt is unknown, as summarized in Table 5. Experiments involving unknown temperatures are described in Appendix F. Specifically, we use GPT-4o-mini (4o-mini) to identify text produced by various LLMs, including GPT-4.1 (4.1), DeepSeek-V3 (DeepSeek), Grok-3-mini (Grok), and Phi-4 (Phi). We also evaluate detection performance on texts generated with different prompts, including versions that have been revised or polished. To create revised text, we use the prompt "*Revise the following text: <human text>.*"; for polished text, we replace "*Revise*" with "*Polish*" in the prompt. The results demonstrates that SearchLLM still achieves the stable performances with unknown models and prompts. SearchLLM achieves comparable results in detecting composite text, as discussed in Appendix G.

Running Time: We evaluate the latency of the detectors as presented in Table 7. When combined with other methods, SearchLLM with RADAR achieves a similar runtime to DetectGPT and requires less than 10 seconds in the worst case.

Attack Scenario: We evaluate the resilience of the detectors under adversarial conditions by employing the DIPPER method (Krishna et al., 2023) with default settings[4] (lexical diversity = 60, order diversity = 0, top p = 0.75) to generate attacks. Specifically, we use DIPPER to target either the original human-written text, the LLM text paraphrased by GPT-4o-mini, or both. This approach allows us to systematically assess how well the detectors can maintain performance when confronted with deliberately altered input designed to evade detection. The empirical results, summarized in Table 8, provide the robustness of the detectors against such adversarial manipulation.

The results demonstrate that DIPPER is capable of mounting successful attacks against all existing methods under evaluation, highlighting a significant vulnerability in current approaches. Notably, the majority of methods experience a remarkable degradation in performance, with a ROC AUC rates dropped when subjected to the DIPPER attack. However, SearchLLM offers a degree of resilience, particularly in scenarios involving LLM-targeted attacks, as it can effectively mitigate the adverse effects imposed by DIPPER.

[4] https://huggingface.co/kalpeshk2011/dipper-paraphraser-xxl

Generation	Para 4o-mini	Para 4.1	Para DeepSeek	Para Grok	Para Phi	Revise 4o-mini	Revise 4.1	Polish 4o-mini
Regeneration	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**
Longformer	0.6552	0.6805	0.4997	0.4996	0.5018	0.6693	0.6761	0.6337
SearchLLM	**0.9073**	**0.8750**	**0.8332**	**0.8342**	**0.8318**	**0.8851**	**0.8528**	**0.8019**
RADAR	0.6543	0.6436	0.6070	0.6038	0.6023	0.6849	0.6761	0.6910
SearchLLM	**0.9062**	**0.8540**	**0.8652**	**0.8646**	**0.8623**	**0.8876**	**0.8433**	**0.8148**
DetectGPT	0.6073	0.5572	0.5664	0.5593	0.5751	0.5993	0.5826	0.6355
SearchLLM	**0.8876**	**0.8396**	**0.8654**	**0.8475**	**0.8700**	**0.8259**	**0.8492**	**0.7917**
Binoculars	0.6388	0.6127	0.4327	0.4351	0.4363	0.6576	0.6670	0.6523
SearchLLM	**0.9070**	**0.8478**	**0.8118**	**0.8112**	**0.8108**	**0.8800**	**0.8492**	**0.8027**

Table 5: Detection of LLM-generated text when the LLMs and prompts are unknown.

Dataset	MAGE (News)	MAGE (QA)	XSum			
Generation	**Topic-based 3.5-turbo**	**Topic-based 3.5-turbo**	**Paraphrase 4o-mini**	**Paraphrase 4o**	**Revise 4o-mini**	**Polish 4o-mini**
Regeneration	**Paraphrase 4o-mini**	**Paraphrase 4o-mini**	**Paraphrase 4o-mini**	**Paraphrase 4o-mini**	**Paraphrase 4o-mini**	**Paraphrase 4o-mini**
Longformer	0.9957	0.9998	0.7179	0.6749	0.7360	0.7440
SearchLLM	**0.9957**	**1.000**	**0.8587**	**0.8257**	**0.8652**	**0.8486**
RADAR	0.9425	0.7992	0.7554	0.7069	0.7728	0.7665
SearchLLM	**0.9591**	**0.8654**	**0.8621**	**0.8194**	**0.8759**	**0.8367**
DetectGPT	0.7079	0.9505	0.4014	0.4424	0.4262	0.4761
SearchLLM	**0.8064**	**0.9639**	**0.6884**	**0.6936**	**0.7016**	**0.6808**
Binoculars	0.9992	0.9940	0.5576	0.5370	0.6112	0.6287
SearchLLM	**0.9995**	**0.9961**	**0.7701**	**0.7387**	**0.8063**	**0.7768**

Table 6: Detection of LLM generated text using Google Search Engine.

Method	w/o Search	with Search
Longformer	3.2s	7.3s
RADAR	0.5s	5.9s
DetectGPT	7.4s	9.4s
Binoculars	6.2s	8.8s

Table 7: Comparison of running times for existing methods with and without SearchLLM(Search) support.

Method	Human	LLM	Both
Longformer	0.5629	0.6069	0.5042
SearchLLM	**0.7644**	**0.7896**	**0.5316**
RADAR	0.5306	0.6484	0.5309
SearchLLM	**0.7686**	**0.7939**	**0.5603**
DetectGPT	0.6305	0.4956	0.4269
SearchLLM	**0.7744**	**0.7230**	**0.4435**
Binoculars	0.2238	0.8676	0.5115
SearchLLM	**0.6121**	**0.9255**	**0.5419**

Table 8: Detection of LLM-generated text manipulated by the DIPPER attack.

4.3 Google Search Scenario

To evaluate detection performance beyond Wikipedia-based texts, we conduct experiments using news-related and QA-related content from the MAGE dataset (Li et al., 2024). Other domains from the RAID dataset (Dugan et al., 2024) and low-resource languages from the M4 dataset (Wang et al., 2024) are presented in Appendix H. Specifically, we leverage the Google Search API to facilitate web-scale analysis but, given the associated high costs, we randomly sample 300 human-written and GPT-3.5-turbo-generated texts from the MAGE dataset. This sample size aligns with the experimental setup employed in the DetectGPT paper (Mitchell et al., 2023).

Furthermore, because the Longformer detector (Li et al., 2024) is trained on MAGE data, we extend our evaluation by randomly selecting 150 human-written articles from the XSum dataset (Narayan et al., 2018). We then generate paraphrased versions of these articles using GPT-4o-mini (4o-mini) and GPT-4o (4o). Additionally, we experiment with revised and polished texts, as shown in Table 6.

The experimental results demonstrate that existing methods perform strongly on the MAGE dataset. However, all existing methods exhibit a significant decline in performance on the XSum dataset, where the LLM-generated text closely mimics human text. SearchLLM improves upon existing methods, particularly with close-mimic texts from the XSum dataset. Moreover, SearchLLM produces more gaps with other methods on both MAGE and XSum dataset for detecting short texts as illustrated in Appendix I.

5 Conclusion

In this paper, we propose a novel method, SearchLLM, for the detection of texts paraphrased by LLMs. SearchLLM operates by first locating potential source material on the internet using a search engine, thereby establishing a pool of candidate texts for comparison. It then assesses the similarity between the input text and a re-generated version derived from these candidate sources to accurately identify LLM-paraphrased content. Experimental results indicate that SearchLLM serves effectively as a proxy to enhance the performance of existing LLM detection methods. Furthermore, our findings suggest that SearchLLM is robust against attacks, particularly those involving paraphrasing techniques intended to evade detection, thereby addressing a key vulnerability in current approaches.

Acknowledgments

We would like to express our sincere gratitude to the anonymous reviewers and area chairs for their valuable comments.

Limitations

Private Source A principal limitation of SearchLLM lies in its reliance on external search engines for information retrieval. Consequently, its scope is inherently restricted to public sources, specifically those that are indexed and accessible via these search engines. This dependency means that SearchLLM is unable to process or retrieve information from private data sources, such as confidential internal documents or personal social media posts authored by individuals.

Freely LLM Text SearchLLM demonstrates limitations when applied to texts that are freely generated by LLMs without being directly based on identifiable sources. Furthermore, when the output of an LLM diverges significantly in meaning or structure from any original source, or when it synthesizes information from sources in a different language, SearchLLM struggles to trace the origins of the content.

Cost SearchLLM utilizes an search engine to retrieve information from the internet and an LLM to regenerate text. Utilizing commercial platforms, such as Google Search or proprietary LLMs like ChatGPT, can incur substantial expenses due to access fees and API usage costs. To mitigate these financial burdens, alternative approaches leverage local or free search engines, such as Wikipedia, and open-source LLMs. While these options typically offer reduced operational costs, they are often constrained by limitations in domain coverage and model capability. Thus, a balance must be struck between cost-effectiveness and the breadth and quality of information retrieval and generation when designing such systems. In particular, for information retrieval, we utilize the Wikipedia API[5] free of charge for queries related to Wikipedia content. For other types of text, SearchLLM issues one query per sample using the Google Search API[6], which incurs a cost of \$0.005 per query. Regeneration is performed only when a suitable candidate is identified. Notably, as shown in Figure 7, approximately 50% of texts yield a valid candidate. With an average of 378.5 tokens per sample, the regeneration cost per sample remains below \$0.00015 from GPT-4o-mini API (\$0.00003 for 200 input tokens and \$0.00012 for 200 output tokens, respectively). Thus, on average, SearchLLM consumes less than \$0.00515 per sample. This is lower than the cost of any GPT models with web search[7], which are priced at \$0.01 per query, as well as other popular models with web search, such as Claude models[8] (\$0.01 per query) and Gemini Flash models[9] (\$0.035 per query).

[5] https://pypi.org/project/wikipedia/
[6] https://developers.google.com/custom-search/v1/overview
[7] https://openai.com/api/pricing/
[8] https://www.claude.com/pricing#api
[9] https://ai.google.dev/gemini-api/docs/pricing

Adaptive Attack The robustness of SearchLLM can be compromised if an attacker intentionally disseminates manipulated text across the internet with the aim of having it indexed by major search engines. This strategy allows the attacker to indirectly inject malicious or misleading content into the retrieval-augmented pipeline of SearchLLM, thereby influencing the model's outputs. To mitigate this vulnerability, it is advisable to constrain search queries to trustworthy sources, such as established news organizations like BBC News. Furthermore, restricting retrieval to textual content published prior to the emergence of LLMs adds an additional safeguard.

References

Guangsheng Bao, Yanbin Zhao, Juncai He, and Yue Zhang. 2025. Glimpse: Enabling white-box methods to use proprietary models for zero-shot llm-generated text detection. In *Proceedings of the 13rd International Conference on Learning Representations (ICLR)*.

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. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

Amrita Bhattacharjee and Huan Liu. 2024. Fighting fire with fire: can chatgpt detect ai-generated text? In *Proceedings of the 30th ACM Conference on Knowledge Discovery and Data Mining Explorations Newsletter (SIGKDD)*, volume 25, pages 14–21.

Yapei Chang, Kalpesh Krishna, Amir Houmansadr, John Wieting, and Mohit Iyyer. 2024. Postmark: A robust blackbox watermark for large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8969–8987.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. Raid: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12463–12492.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. 2025. Black-box detection of language model watermarks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hanxi Guo, Siyuan Cheng, Xiaolong Jin, Zhuo Zhang, Kaiyuan Zhang, Guanhong Tao, Guangyu Shen, and Xiangyu Zhang. 2024a. Biscope: Ai-generated text detection by checking memorization of preceding tokens. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 37, pages 104065–104090.

Xun Guo, Yongxin He, Shan Zhang, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024b. Detective: Detecting ai-generated text via multi-level contrastive learning. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 37, pages 88320–88347.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 17519–17537.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pages 15077–15095.

Mingjia Huo, Sai Ashish Somayajula, Youwei Liang, Ruisi Zhang, Farinaz Koushanfar, and Pengtao Xie. 2024. Token-specific watermarking with enhanced detectability and semantic coherence for large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 20746–20767.

Zae Myung Kim, Kwang Hee Lee, Preston Zhu, Vipul Raheja, and Dongyeop Kang. 2024. Threads of subtlety: Detecting machine-generated texts through discourse motifs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5449–5474.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 17061–17084.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, volume 38, pages 21258–21266.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pages 27469–27500.

Xiang Li, Zhiyi Yin, Hexiang Tan, Shaoling Jing, Du Su, Yi Cheng, Huawei Shen, and Fei Sun. 2025. Prdetect: Perturbation-robust llm-generated text detection based on syntax tree. In *Findings of the Association for Computational Linguistics (NAACL)*, pages 8290–8301.

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 (ACL)*, pages 36–53.

Shengchao Liu, Xiaoming Liu, Yichen Wang, Zehua Cheng, Chengzhengxu Li, Zhaohan Zhang, Yu Lan, and Chao Shen. 2024. Does detectgpt fully utilize perturbation? bridging selective perturbation to fine-tuned contrastive learning detector would be better. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1874–1889.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2023. Coco: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 16167–16188.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. 2024. An entropy-based text watermarking detection method. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11724–11735.

Shixuan Ma and Quan Wang. 2024. Zero-shot detection of llm-generated text using token cohesiveness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 17538–17553.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative ai detection via rewriting. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

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 *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 278–293.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 24950–24962. PMLR.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1797–1807.

Hoang-Quoc Nguyen-Son, Minh-Son Dao, and Koji Zettsu. 2024. Simllm: Detecting sentences generated by large language models using similarity between the generation and its re-generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 22340–22352.

Hyeonchu Park, Byungjun Kim, and Bugeun Kim. 2025. Dart: An aigt detector using amr of rephrased text. In *Proceedings of the Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, pages 710–721.

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. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 494–502.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 12395–12412.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. 2024. Multiscale positive-unlabeled detection of ai-generated texts. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1702–1717.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, and 1 others. 2024. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (ACL)*, pages 1369–1407.

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)*, pages 2113–2133.

Yang Xu, Yu Wang, Hao An, Zhichen Liu, and Yongyuan Li. 2024. Detecting subtle differences between human and model languages using spectrum of relative likelihood. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. 2025. Training-free llm-generated text detection by mining token probability sequences. In *Proceedings of the 13rd International Conference on Learning Representations (ICLR)*.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

Xiao Yu, Kejiang Chen, Qi Yang, Weiming Zhang, and Nenghai Yu. 2024a. Text fluoroscopy: Detecting llm-generated text through intrinsic features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 15838–15846.

Xiao Yu, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Xiuwei Shang, Weiming Zhang, and Nenghai Yu. 2024b. Dpic: Decoupling prompt and intrinsic characteristics for llm generated text detection. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 37, pages 16194–16212.

Cong Zeng, Shengkun Tang, Xianjun Yang, Yuanzhou Chen, Yiyou Sun, Zhiqiang Xu, Yao Li, Haifeng Chen, Wei Cheng, and Dongkuan DK Xu. 2024. Dald: Improving logits-based detector without logits from black-box llms. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 37, pages 54947–54973.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023. Beat llms at their own game: Zero-shot llm-generated text detection via querying chatgpt. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7470–7483.

A Algorithms

The processes of SearchLLM for detecting LLM-generated text and matching input sentences with URL content are described in Algorithm 1 and Algorithm 2, respectively.

B Other Metrics

As shown in Table 9, we report the F-score from the same experiments described in Table 1. The results demonstrate that the F-score yields outcomes similar to the ROC AUC score. We also report the ROC AUC at an FPR of 1% for other main tables in this paper, as shown in Tables 10–13. SearchLLM demonstrates more stable scores than Binoculars and achieves a higher average score overall. In the other tables, SearchLLM achieves ROC AUC scores that are greater than or equal to those of other existing methods.

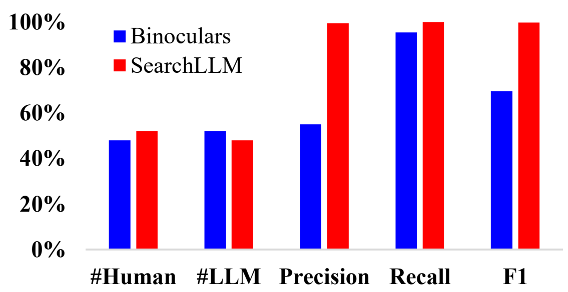

Figure 9: The number of human- and LLM-generated texts processed by SearchLLM and Binoculars, along with their performance in detecting GPT-4.1-generated content.

C Statistical Test

In accordance with the experiment presented in Table 3, we conduct paired t-tests to compare the performance of existing methods with and without the support of SearchLLM. The results, shown in Table 14, demonstrate that all methods supported by SearchLLM perform significantly better than those without support, with all p-values less than 0.004.

D Breakdown of Performance on Detecting GPT-4.1 Generated Text

We provide a detailed analysis of the performance of SearchLLM combined with Binoculars in detecting GPT-4.1 generated text, as shown in Figure 9. SearchLLM and Binoculars demonstrate a similar trend when detecting GPT-4o-mini generated text, as illustrated in Figure 7.

E Change of Other Parameters

In addition to the human threshold α discussed in Figure 8, we also evaluate the effects of varying other thresholds, including the machine threshold β (Figure 10), the regeneration threshold Δ (Figure 11), and the ratio threshold γ (Figure 12). The results indicate that the optimal values for β, Δ, and γ are relatively high, low, and middle ranges, respectively.

F Various Temperature Settings

We conduct experiments to evaluate the detection of LLM-paraphrased text generated by GPT-4o-

Algorithm 1: SearchLLM for LLM text detection.

```
Input  : Input text t
Output : Original/Generated, Confidence score
U ← SEARCH(t)                                              ▷ List of urls
for each u_i in U do
    t_c ← EXTRACT_CANDIDATE(u_i, t)
    if t_c ≠ None then
        σ_c ← MEASURE_SIMILARITY(t_c, t)
        if σ_c ≥ α then
            return Original, σ_c
        else
            t_r ← REGENERATE(t_c)
            σ_r ← MEASURE_SIMILARITY(t_r, t)
            if σ_r − σ_c ≥ Δ then
                return Generated, σ_r − σ_c
            else
                return DETECT_BY_AN_EXISTING_METHOD(t)
            end if
        end if
    end if
end for
return DETECT_BY_AN_EXISTING_METHOD(t)
```

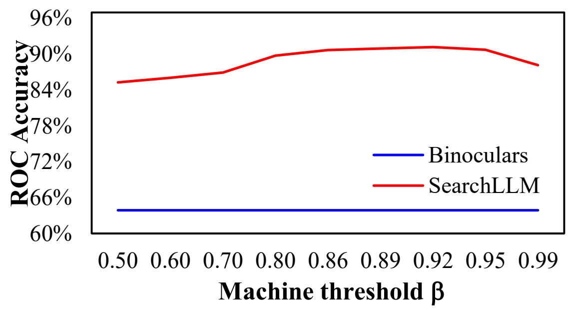

Figure 10: The impact of varying the machine threshold parameter β on the performance of SearchLLM.

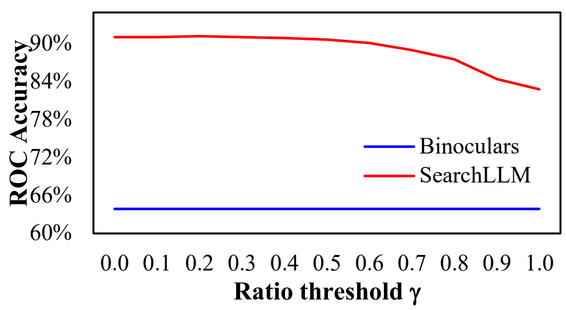

Figure 12: The impact of varying the ratio threshold parameter γ on the performance of SearchLLM.

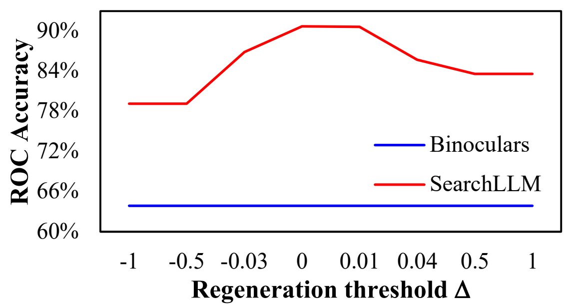

Figure 11: The impact of varying the regeneration threshold parameter Δ on the performance of SearchLLM.

mini at various temperature settings, as shown in Figure 13 and Figure 14. The results indicate that Binoculars is more sensitive to temperature changes than RADAR. Meanwhile, SearchLLM consistently demonstrates improved performance across different temperatures.

G Composite

We evaluate the ability to detect composite text by dividing each sample into equal parts. The first part is either written by a human or paraphrased by GPT-4o-mini, while the second part is paraphrased or revised by GPT-4o-mini. The results, shown in Table 15, demonstrate that SearchLLM outperforms existing methods across all combinations.

Algorithm 2: Matching input sentences with URL content.

Input : Input text t, URL u
Output : Matched pair P

$S_t \leftarrow$ SPLIT_SENTENCE(t)
$S_u \leftarrow$ SPLIT_SENTENCE(u)
$b \leftarrow \emptyset$ ▷ Best index
for *each* S_{u_i} *in* S_u **do**
  if $b == \emptyset$ *or* $\mathrm{SIM}(S_{t_1}, S_{u_i}) > \mathrm{SIM}(S_{t_1}, S_{u_b})$ **then**
    $b \leftarrow i$
  end if
end for
$S_u \leftarrow S_u \setminus \{s_1, s_2, \ldots, s_{b-1}\}$ ▷ Update S_u by removing its first $(b-1)$ sentences.
$P \leftarrow \emptyset$ ▷ Matched pairs
while $S_u \neq \emptyset$ *and* $S_t \neq \emptyset$ **do**
  $p \leftarrow \emptyset$ ▷ Best current pair
  for k *from* 1 *to* $len(S_u)$ **do**
    $C_k = S_u|_{1:k}$ ▷ the first k sentences of S_u
    if $p = 0$ *or* $\mathrm{SIM}(S_{t_1}, C_k) > \mathrm{SIM}(p)$ **then**
      $p \leftarrow \{S_{t_1}, C_k\}$
    end if
  end for
  for l *from* 1 *to* $len(S_t)$ **do**
    $C_l = S_t|_{1:l}$ ▷ the first l sentences of S_t
    if $\mathrm{SIM}(C_l, S_{u_1}) > \mathrm{SIM}(p)$ **then**
      $p \leftarrow \{C_l, S_{u_1}\}$
    end if
  end for
  $P \leftarrow P \cup \{p\}$ ▷ Add p to P
  $S_t \leftarrow S_t \setminus p$ ▷ Update S_t by removing p from S_t
  $S_u \leftarrow S_u \setminus p$ ▷ Update S_u by removing p from S_u
end while
return P

Method	GPT-4	Llama-2-70B-Chat	Cohere-Chat	MPT-30B-Chat	Mistral-7B-Chat
Longformer	0.9705	0.9879	0.8827	0.9609	0.9592
SearchLLM	**0.9802**	**0.9943**	**0.9195**	**0.9768**	**0.9731**
RADAR	0.9605	0.9794	0.8958	0.9759	0.9712
SearchLLM	**0.9702**	**0.9866**	**0.9249**	**0.9832**	**0.9796**
DetectGPT	0.8298	0.8683	0.7383	0.8031	0.8305
SearchLLM	**0.8867**	**0.9143**	**0.8418**	**0.8781**	**0.8856**
Binoculars	0.9912	**0.9997**	0.9847	**0.9997**	0.9985
SearchLLM	**0.9935**	**0.9997**	**0.9893**	**0.9997**	**0.9985**

Table 9: Performance of LLM-generated text detection on all Wikipedia-related samples from the RAID dataset (F-score).

H Google Search Scenario with Other Datasets

We evaluate the capability of our method in detecting all other domains within the RAID dataset (Dugan et al., 2024), including News, Book, Abstract, Poetry, Recipe, Reddit, and Reviews, as shown in Table 16. SearchLLM consistently achieves a ROC AUC greater than 0.8 across all

Method	GPT-4o-mini	GPT-4.1	DeepSeek-V3	Grok-3-mini	Phi-4
Longformer	0.0076	0.0099	0.0024	0.0015	0.0025
SearchLLM	**0.4231**	**0.3977**	**0.3930**	**0.3079**	**0.3530**
RADAR	0.0136	0.0132	0.0229	0.0217	0.0184
SearchLLM	**0.4173**	**0.3594**	**0.4178**	**0.3225**	**0.3660**
DetectGPT	0.0065	0.0062	0.0049	0.0092	0.0046
SearchLLM	**0.4122**	**0.3820**	**0.3879**	**0.3467**	**0.3354**
Binoculars	0.0115	0.0168	0.0090	0.0113	0.0039
SearchLLM	**0.4188**	**0.3890**	**0.3989**	**0.3592**	**0.3321**

Table 10: Detection of paraphrased text generated by various large language models (ROC AUC at an FPR of 1%).

Generation	Para 4o-mini	Para 4.1	Para DeepSeek	Para Grok	Para Phi	Revise 4o-mini	Revise 4.1	Polish 4o-mini
Regeneration	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**	**Para 4o-mini**
Longformer	0.0076	0.0099	0.0024	0.0015	0.0025	0.0089	0.0058	0.0048
SearchLLM	**0.4231**	**0.2981**	**0.2649**	**0.2603**	**0.2603**	**0.3355**	**0.1947**	**0.0957**
RADAR	0.0136	0.0132	0.0229	0.0217	0.0184	0.0107	0.0137	0.0150
SearchLLM	**0.4173**	**0.3087**	**0.2906**	**0.2848**	**0.2834**	**0.3555**	**0.2314**	**0.1172**
DetectGPT	0.0065	0.0088	0.0088	0.0044	0.0044	0.0044	0.0044	0.0133
SearchLLM	**0.4122**	**0.8542**	**0.2044**	**0.1688**	**0.1911**	**0.2977**	**0.1911**	**0.0800**
Binoculars	0.0115	0.0168	0.0090	0.0113	0.0039	0.0228	0.0251	0.0221
SearchLLM	**0.4188**	**0.3369**	**0.2545**	**0.2670**	**0.2604**	**0.4137**	**0.2526**	**0.1329**

Table 11: Detection of LLM-generated text when the LLMs and prompts are unknown (ROC AUC at an FPR of 1%).

Method	Human	LLM	Both
Longformer	0.0075	0.0064	0.0051
SearchLLM	**0.3723**	**0.0687**	**0.0087**
RADAR	0.0000	0.0376	0.0032
SearchLLM	**0.3723**	**0.1043**	**0.0087**
DetectGPT	0.0177	**0.0000**	0.0044
SearchLLM	**0.3155**	**0.0000**	**0.0311**
Binoculars	0.0008	0.1504	0.0037
SearchLLM	**0.3723**	**0.2565**	**0.0087**

Table 12: Detection of LLM-generated text manipulated by the DIPPER attack (ROC AUC at an FPR of 1%).

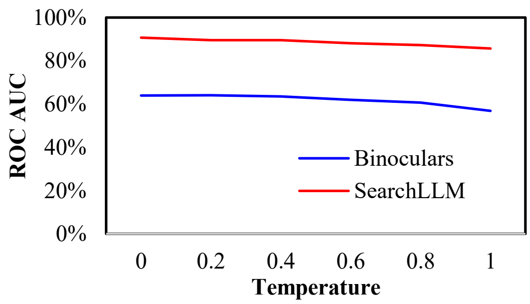

Figure 13: Performance comparison of Binoculars and SearchLLM in detecting LLM-generated text across different temperature settings.

domains, outperforming DetectGPT and matching or exceeding the performance of other existing methods.

Furthermore, we extend our experiments to detect content in all low- and medium-resource languages in the M4 dataset (Wang et al., 2024), including Urdu (low-resource), Bulgarian (low-resource), and Indonesian (medium-resource), as presented in Table 17. The results demonstrate that SearchLLM efficiently improves the performance compared to all existing methods.

I Google Search Scenario with Short Text

In alignment with the experiments listed in Table 6, we report the results of detecting LLM-generated text on short texts limited to 30 words, as shown in Table 18. The results demonstrate that SearchLLM generally exhibits larger performance gaps compared to other methods in both the MAGE and XSum experiments.

Dataset	MAGE (News)	MAGE (QA)	XSum			
Generation	Topic-based **3.5-turbo**	Topic-based **3.5-turbo**	Paraphrase **4o-mini**	Paraphrase **4o**	Revise **4o-mini**	Polish **4o-mini**
Regeneration	Paraphrase **4o-mini**	Paraphrase **4o-mini**	Paraphrase **4o-mini**	Paraphrase **4o-mini**	Paraphrase **4o-mini**	Paraphrase **4o-mini**
Longformer	**0.0000**	0.0466	0.0155	0.0155	0.0222	0.0244
SearchLLM	**0.0000**	**0.3822**	**0.1177**	**0.1177**	**0.1644**	**0.0822**
RADAR	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0088	0.0088
SearchLLM	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.1600**	**0.0666**
DetectGPT	**0.0000**	**0.2088**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
SearchLLM	**0.0000**	**0.2088**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Binoculars	0.6488	**0.6533**	0.0044	0.0044	0.0533	0.0266
SearchLLM	**0.6577**	**0.6533**	**0.1066**	**0.1066**	**0.2044**	**0.0844**

Table 13: Detection of LLM generated text using Google Search Engine (ROC AUC at an FPR of 1%).

Method	GPT-4o-mini	GPT-4.1	DeepSeek-V3	Grok-3-mini	Phi-4
SearchLLM vs Longformer	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
SearchLLM vs RADAR	0.0033	< 0.0001	< 0.0001	< 0.0001	< 0.0001
SearchLLM vs DetectGPT	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
SearchLLM vs Binoculars	0.0020	< 0.0001	< 0.0001	< 0.0001	< 0.0001

Table 14: p-values from paired t-tests assessing the detection of LLM-paraphrased text generated by various language models.

Method	Human and Paraphrase	Human and Revise	Paraphrase and Revise
Longformer	0.5660	0.5850	0.6316
SearchLLM	**0.7866**	**0.7907**	**0.8870**
RADAR	0.5846	0.6082	0.6671
SearchLLM	**0.7781**	**0.7800**	**0.8984**
DetectGPT	0.5499	0.5816	0.6109
SearchLLM	**0.7672**	**0.7694**	**0.8723**
Binoculars	0.5667	0.5811	0.6389
SearchLLM	**0.7862**	**0.7821**	**0.8866**

Table 15: Detection of composite text generated by large language models.

Method	News	Books	Abstract	Poetry	Recipe	Reddit	Reviews
Longformer	0.9934	0.9987	**1.0000**	0.9588	0.8332	0.9914	**1.0000**
SearchLLM	**0.9999**	**0.9994**	**1.0000**	**0.9817**	**0.8499**	**0.9955**	**1.0000**
RADAR	**1.0000**	0.9995	0.9946	0.8462	0.9973	0.9841	0.9948
SearchLLM	**1.0000**	**0.9996**	**1.0000**	**0.9078**	**0.9973**	**0.9942**	**1.0000**
DetectGPT	0.7861	0.9292	0.8830	0.7475	0.9519	0.9561	0.9387
SearchLLM	**0.9180**	**0.9825**	**0.9956**	**0.9451**	**0.9624**	**0.9666**	**0.9721**
Binoculars	**1.0000**	**1.0000**	**1.0000**	0.9999	**0.9999**	0.9984	0.9999
SearchLLM	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9999**	**0.9992**	**1.0000**

Table 16: Detection of LLM-generated text across all other domains in the RAID dataset using the Google Search Engine.

Method	Urdu	Bulgarian	Indonesian
Longformer	0.4393	0.7703	0.7323
SearchLLM	**0.5817**	**0.8829**	**0.8161**
RADAR	0.4858	0.6715	0.6855
SearchLLM	**0.6069**	**0.8459**	**0.7873**
DetectGPT	0.4371	0.5474	0.5116
SearchLLM	**0.5707**	**0.8122**	**0.6900**
Binoculars	0.7968	0.7891	0.9543
SearchLLM	**0.8386**	**0.8786**	**0.9598**

Table 17: Detection of LLM-generated text in the M4 dataset using Google Search for low- and medium-resource languages.

Dataset	MAGE (News)	MAGE (QA)	XSum			
Generation	Topic-based **3.5-turbo**	Topic-based **3.5-turbo**	Paraphrase **4o-mini**	Paraphrase **4o**	Revise **4o-mini**	Polish **4o-mini**
Regeneration	Paraphrase **4o-mini**	Paraphrase **4o-mini**	Paraphrase **4o-mini**	Paraphrase **4o-mini**	Paraphrase **4o-mini**	Paraphrase **4o-mini**
Longformer	0.8944	0.9345	0.4270	0.4505	0.4496	0.4840
SearchLLM	**0.9348**	**0.9607**	**0.7197**	**0.7369**	**0.7378**	**0.6934**
RADAR	0.7879	0.5624	0.6198	0.6387	0.5983	0.5974
SearchLLM	**0.8651**	**0.6950**	**0.8099**	**0.8234**	**0.7998**	**0.7461**
DetectGPT	0.6760	0.8204	0.4431	0.4254	0.4429	0.4776
SearchLLM	**0.8136**	**0.8915**	**0.7208**	**0.7180**	**0.7218**	**0.6739**
Binoculars	0.8395	0.9570	0.5707	0.5401	0.6009	0.5785
SearchLLM	**0.9234**	**0.9747**	**0.7875**	**0.7656**	**0.8052**	**0.7216**

Table 18: Detection of LLM-generated short text using the google search engine.

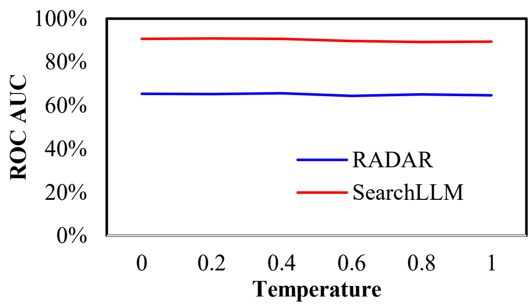

Figure 14: Performance comparison of RADAR and SearchLLM in detecting LLM-generated text across different temperature settings.