

# PlagBench: Exploring the Duality of Large Language Models in Plagiarism Generation and Detection

Jooyoung Lee<sup>1</sup> Toshini Agrawal<sup>2</sup> Adaku Uchendu<sup>3</sup>  
Thai Le<sup>4</sup> Jinghui Chen<sup>1</sup> Dongwon Lee<sup>1</sup>

<sup>1</sup> Penn State University, PA, USA, {jfl15838, jzc5917, dongwon}@psu.edu

<sup>2</sup> Vellore Institute of Technology, Bhopal, India, agrawaltoshini@gmail.com

<sup>3</sup> MIT Lincoln Laboratory, MA, USA, adaku.uchendu@ll.mit.edu

<sup>4</sup> Indiana University, IN, USA, tle@iu.edu

## Abstract

Recent studies have raised concerns about the potential threats large language models (LLMs) pose to academic integrity and copyright protection. Yet, their investigation is predominantly focused on literal copies of original texts. Also, how LLMs can facilitate the detection of LLM-generated plagiarism remains largely unexplored. To address these gaps, we introduce PlagBench, a dataset of 46.5K synthetic text pairs that represent three major types of plagiarism: verbatim copying, paraphrasing, and summarization. These samples are generated by three advanced LLMs. We rigorously validate the quality of PlagBench through a combination of fine-grained automatic evaluation and human annotation. We then utilize this dataset for two purposes: (1) to examine LLMs’ ability to transform original content into accurate paraphrases and summaries, and (2) to evaluate the plagiarism detection performance of five modern LLMs alongside three specialized plagiarism checkers. Our results show that GPT-3.5 Turbo can produce high-quality paraphrases and summaries without significantly increasing text complexity compared to GPT-4 Turbo. However, in terms of detection, GPT-4 outperforms other LLMs and commercial detection tools by 20%, highlights the evolving capabilities of LLMs not only in content generation but also in plagiarism detection. Data and source code are available at <https://github.com/Brit7777/plagbench>.

## 1 Introduction

Plagiarism occurs when someone uses another person’s work, ideas, or expressions without proper acknowledgment. It is one of the most frequently scrutinized issues in writing tasks, which leads to a violation of intellectual property rights and academic integrity. Traditionally, plagiarism research has focused on human-written text. However, the rise of large language models (LLMs) has transformed the landscape of plagiarism (Pudasaini

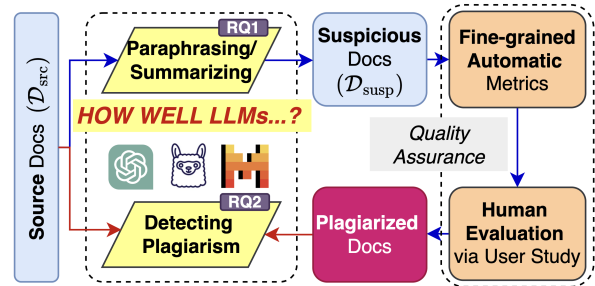


Figure 1: The overview of PlagBench construction processes and proposed RQs. Blue and red arrows denote the flow of RQ1 and RQ2, respectively.

et al., 2024). For instance, users can exploit these models to obfuscate original content through paraphrases or summaries, effectively bypassing traditional plagiarism detection systems (Krishna et al., 2024; Sadasivan et al., 2023; Weber-Wulff et al., 2023; Elkhatat et al., 2021). Moreover, many recent studies (Carlini et al., 2021; Tirumala et al., 2022; Zhang et al., 2023; Carlini et al., 2023) have shown that LLMs can memorize portions of their training data and reproduce them during text generation. These models are not limited to producing verbatim copies; they can also generate paraphrased or elongated versions of unique content (Lee et al., 2023b).

The increasing prevalence of plagiarism in LLMs raises two critical research questions: (RQ1) *how well can LLMs generate paraphrase and summary plagiarism of given texts?* and (RQ2) *how well can LLMs detect three types of plagiarism—i.e., verbatim, paraphrase, and summary plagiarisms?* While prior research has explored LLMs’ applications to automated paraphrase (Witteveen and Andrews, 2019; Wahle et al., 2022) and summary (Lingard, 2023; Goyal et al., 2022; Liu et al., 2023) generation, these studies are often limited in scope because most of them focus on individual domains or specific models. A holistic and fine-grained analysis of diverse domains and models is crucial to

developing robust mechanisms for detecting and mitigating plagiarism in AI-generated content. Additionally, LLMs have demonstrated impressive performance in various natural language processing (NLP) tasks (Chang et al., 2024), motivating us to investigate their potential in automatic plagiarism detection. Current plagiarism detectors are primarily optimized to catch literal copies or superficial paraphrasing by humans and may not be sufficient to detect nuanced rewritings of LLMs. Evaluating LLMs’ effectiveness in plagiarism detection can offer critical insights into the limitations of current systems and help guide the development of more advanced, robust solutions.

To answer these questions, we present **PlagBench**, a large-scale corpus containing 46.5K text pairs with three plagiarism categories: verbatim, paraphrase, and summary plagiarism. These samples cover three writing domains (i.e., abstract of scholarly paper, story, news article) and are synthetically generated by three advanced, closed-source and open-source LLMs (Llama2-70b-chat (Touvron et al., 2023), GPT-3.5 Turbo<sup>1</sup>, and GPT-4 Turbo (Achiam et al., 2023)). To acquire the gold standard for automatic plagiarism detection tasks, we employ a comprehensive integration of both automatic and manual evaluation. This final subset is then used for evaluation of five popular LLMs (Llama2-70b-chat, Llama3-70b-instruct<sup>2</sup>, Mixtral-8x7B (Jiang et al., 2024), GPT-3.5 Turbo, and GPT-4 Turbo) and three specialized plagiarism detectors (GPTZero<sup>3</sup>, Prepostseo<sup>4</sup>, and a detector proposed by Lee et al. (2023b)), two of which are commercial-ready. Figure 1 illustrates a visual representation of our dataset construction processes and the proposed RQs.

Key contributions of this work are as follows: (1) **PlagBench** is the first dataset that covers a wide range of plagiarism examples generated by state-of-the-art LLMs; (2) we showcase the strengths and weaknesses of LLM-generated plagiarism cases across domains; (3) our results show the potential of LLMs facilitating plagiarism detection, although they generally have difficulties in distinguishing the type of summary plagiarism.

<sup>1</sup><https://platform.openai.com/docs/models/gpt-3-5-turbo>

<sup>2</sup><https://ai.meta.com/blog/meta-llama-3/>

<sup>3</sup><https://gptzero.me/plagiarism-checker>

<sup>4</sup><https://www.prepostseo.com/plagiarism-checker>

## 2 Related Work

### 2.1 Plagiarism Detection Corpus

The Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005) and PAWS (Zhang et al., 2019b) are among the most renowned datasets for paraphrase identification. Yet, they are not suitable for plagiarism detection tasks because they only cover one type. Additionally, their samples are annotated at the sentence level, which limits their applicability in real-world scenarios where plagiarism in general occurs at the paragraph level. More recently, Wahle et al. (2021) constructed a paragraph-long machine-paraphrased plagiarism dataset using three neural language models such as BERT (Devlin et al., 2019a), RoBERTa (Liu, 2019), and Longformer (Beltagy et al., 2020). Following, the authors replaced these three models with more recently released LLMs, GPT-3, and T5, and produced machine-paraphrased texts (Wahle et al., 2022). There are many independent machine-summarized text pairs (e.g., Cohan et al. (2018); Narayan et al. (2018)) as well, but there exists no comprehensive dataset covering both paraphrases and summaries generated by modern LLMs.

Potthast et al. (2013) introduced a novel plagiarism detection dataset with five obfuscation strategies: no obfuscation, random obfuscation, translation obfuscation, and summary obfuscation. These strategies utilized tools like Google Translate and text summarizers to mimic plagiarist behaviors. To the best of our knowledge, this is the first and only plagiarism detection dataset covering a wide range of plagiarism types. However, this dataset has not been updated since 2013.

### 2.2 Automatic Plagiarism Detection

There are two formats of the automated plagiarism detection task: *intrinsic* and *extrinsic* plagiarism detection. While intrinsic plagiarism detection analyzes the document itself without resorting to the reference document, extrinsic plagiarism detection involves directly comparing the suspicious document against the reference to detect plagiarism instances (Alzahrani et al., 2011). Intrinsic approaches use patterns such as linguistic differences within a document which indicates multiple authorship to detect plagiarism (Gipp and Gipp, 2014; Xiong et al., 2024; Potthast et al., 2010; Song et al., 2024; Stein et al., 2011). Due to the high complexity of intrinsic plagiarism tasks, many popular detection tools follow extrinsic plagiarism

detection strategies. The extrinsic plagiarism detection task is generally divided into two subtasks: source retrieval (i.e., finding the most plausible pair of the source and the suspicious document) and plagiarism identification (i.e., classifying whether two documents are plagiarizing each other or not). These approaches rely on similarity scores of two non-zero vectors transformed from the document pairs and apply certain thresholds (Vani and Gupta, 2017; Sağlam et al., 2024b; Potthast et al., 2010; Sağlam et al., 2024a; Lee et al., 2023a; Moravvej et al., 2023; Avetisyan et al., 2023). Our focus lies in the latter scope. We assume we already have the text pair, the source document, and the suspicious document obtained through PlagBench, and we concentrate on plagiarism identification.

### 3 PlagBench Corpus Generation

#### 3.1 Corpus Construction

**Source Document Collection.** Although plagiarism can occur in any type of writing, we target specific areas such as scholarly or creative works, where originality and intellectual property are highly valued. Li et al. (2023) present a collection of human-authored and machine-authored corpora from diverse writing tasks. Among their selections of corpora, we carefully choose three public English datasets: (1) **SciXGen** (Chen et al., 2021) consisting of 200K+ abstracts of scientific articles (*for scientific writing*), (2) **ROCStories** (Mostafazadeh et al., 2016) consisting of 50k five-sentence commonsense stories (*for story writing*); and (3) **TLDR**<sup>5</sup> consisting of 7K+ TLDR tech newsletters (*for news article writing*). Their samples contain one human-written text and multiple machine-generated variations, all sharing either the same topic or text source as the human-written piece. For the purpose of the study, we only use the human-written text containing at least 5 sentences as the source text.<sup>6</sup> We randomly select 2,400 samples from SciXGen and ROCStories corpora and 1,300 from TLDR for suspicious document generation. The sample size of TLDR is smaller than the other two datasets because the content of TLDR tends to be short.

**Suspicious Document Generation.** Instruction-tuned LLMs have demonstrated strong abilities in

text generation that are aligned with users’ natural language commands (Lingard, 2023; Zhang et al., 2024; Tang et al., 2023). Hence, we employ Llama2-70b-chat, GPT-3.5 Turbo, and GPT-4 Turbo models to generate texts corresponding to two plagiarism labels (i.e., paraphrase and summary plagiarism), respectively. We do not use these models for verbatim plagiarism, as generating suspicious documents can be easily accomplished by simple copy-and-paste without any modification. Table 8 shows our hand-crafted prompt templates, each tailored to transform a single source document according to the provided task descriptions.

In plagiarism-free scenarios, existing datasets (e.g., Potthast et al. (2013), Foltýnek et al. (2020)) pair random or dissimilar documents. However, this often results in pairs that are clearly different and thus easily identifiable. More challenging pairs are likely to occur when two documents share the same key topics and domain type, but their details do not overlap. Consequently, we instruct LLMs to write continuations based on provided keywords, domain information, and the first two sentences of the source document. Two meaningful keywords are automatically extracted using KeyBERT,<sup>7</sup> a package leveraging contextual embeddings from BERT (Devlin et al., 2019b).

**Generation Results.** We have in total 6,100 source documents across three writing domains. Using three LLMs, we create three corresponding paraphrase, summary, and plagiarism-free documents per sample. For GPT-4, due to its relatively high API cost<sup>8</sup>, we only utilize a subset (n=3,300) of source documents for generation. We use the version of *gpt-3.5-turbo-1106* and *gpt-4-1106-preview* and apply temperature sampling ( $t = 0.9$ ) for the generation configuration. As a result, the total number of generated suspicious documents is 46,500, a summation of 6,100 generations for SciXGen, 6,100 generations for ROCStories, and 3,300 generations for TLDR, multiplied by 3 plagiarism types.

#### 3.2 Corpus Quality Assurance via Automated Metrics

**Automatic Evaluation.** Hallucination in LLMs is a widely recognized problem. To ensure that their generations are accurate paraphrases and sum-

<sup>5</sup>[https://huggingface.co/datasets/JulesBelveze/tldr\\_news](https://huggingface.co/datasets/JulesBelveze/tldr_news)

<sup>6</sup>According to our pilot experiment, paraphrases and summaries generated based on documents shorter than 5 sentences tend to be too similar to distinguish even from the human lens.

<sup>7</sup><https://maartengr.github.io/KeyBERT/index.html>

<sup>8</sup>Llama2 families are open-source models while GPT-3.5 turbo and GPT-4 Turbo charge \$3.00 and \$40.00 per 1M tokens respectively.

Type	Metric	Description
Paraphrase	Semantic equivalence	Evaluates whether the paraphrased text conveys the same meaning as the source text.
	Consistency	Evaluates whether the paraphrased text accurately represents the information contained in the source text without the inclusion of errors or distortions.
	Language quality	Evaluates whether the paraphrased text maintains or improves upon the quality of the source text in regards to fluency and grammaticality.
Summary	Relevance	Evaluates whether the summary covers all the essential information and key points from the source text while omitting irrelevant or tangential details.
	Coherence	Evaluate whether the summary is well-structured and not just a random assortment of information from the source text.
	Consistency	Evaluates whether the summary accurately represents the content and meaning of the source text without the inclusion of errors or distortions.
	Language quality	Evaluates whether the summary maintains or improves upon the quality of the source text in regards to fluency and grammaticality.

Table 1: Descriptions of evaluation metrics for paraphrase and summary plagiarism.

maries, we rigorously evaluate them based on our established metrics (Table 1) and apply filtering strategies. Due to relatively large volumes of generated samples, we first conduct automatic measurements for paraphrase and summary plagiarism. Note that we do not perform automatic evaluation for non-plagiarism cases because it would overlap with our benchmark task and necessitate the use of the same plagiarism detectors we aim to evaluate.

For paraphrase plagiarism, we perform an evaluation of LLM-paraphrased documents through the lens of semantic equivalence, consistency, and language quality. Several papers (Shen et al., 2022; Chen and Dolan, 2011) adopted similar criteria for paraphrase evaluation. We use the source documents as reference documents and carefully choose a single representative metric for every aspect.

- **Semantic equivalence** → BERTScore (Zhang et al., 2019a) : BERTScore retrieves the token-level embedding using BERT and computes the summation of cosine similarities. It is shown to be more robust for paraphrase classification than other conventional metrics like BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005).
- **Consistency** → AlignScore (Zha et al., 2023) : the authors train a information alignment model on 7 well-established tasks including paraphrasing and summarization.
- **Language quality** → Readability index (Senter and Smith, 1967) & CoLAScore (Zhu and Bhat, 2020) : we use automated readability index for fluency and use COLAScore, a RoBERTa-based model finetuned on grammaticality measurement

data, for grammaticality measurement.

Now we turn our attention to automatic summary evaluation. Specifically, we measure relevance, coherence, consistency, and language quality, suggested by the SummEval benchmark (Fabbri et al., 2021). In the absence of gold-standard summaries for comparison with LLM-generated summaries, we rely on reference-free metrics. Below are the descriptions of selected metrics:

- **Relevance** → BLANC (Vasilyev et al., 2020) : BLANC is a novel approach that calculates the usefulness of a summary in helping BERT for language understanding task. This metric is ranked the highest in relevance alignment in SummEval.
- **Coherence** → BARTScore (Yuan et al., 2021) : BARTScore is a comprehensive evaluation metric using a pre-trained sequence-to-sequence (seq2seq) model BART (Lewis et al., 2020).
- We use the same models from paraphrase evaluation regarding consistency and language quality.

Most metrics, except for CoLAScore, provide continuous values, requiring threshold selection for filtering. The primary goal of automated evaluation is to eliminate inaccurate pairs. Thus, we shuffle the order of suspicious document rows while keeping the source document order intact and create invalid pairs as a means of establishing a loose cut-off point. Refer to Appendix A.1 for experiment configurations and the threshold setup. Lastly, we remove near-duplicates based on the Levenshtein distance (Miller et al., 2009) between two texts.

**Automatic Filtering Results.** We remove samples from a final set if the text pair does not satisfy all 3 or 4 aspects, depending on the plagiarism category. Among 31,000 (15,500 per plagiarism type) suspicious documents, 12,071 samples for paraphrase plagiarism and 13,445 samples for summary plagiarism remain intact. The document percentage breakdown per plagiarism type and domain type after automatic filtering is illustrated in Table 6.

### 3.3 Corpus Quality Assurance via Manual Annotation

Human annotation plays a critical role in verifying the accuracy and reliability of automated systems. We use Amazon Mechanical Turk (AMT) to hire human annotators for three separate annotation tasks: non-plagiarism evaluation, paraphrase evaluation, and summary evaluation. See Appendix A.2 for more details on the experiment design. Due to limited budgets, we perform manual annotation on 459 batches (i.e., 1,377 samples) for no plagiarism, 393 batches (i.e., 1,251 samples) for paraphrase plagiarism, and 417 batches (i.e., 1,251 samples) for summary plagiarism cases that are randomly sampled from §3.2.

We take several actions to rigorously reinforce annotation qualities. First, we employ four built-in worker qualifications, including (1) HIT Approval Rate of  $\leq 98\%$ , (2) a minimum of 1,000 Approved HITs, (3) Masters qualification status, and (4) U.S.-based workers. Second, a honey pot question is included to detect automated responses or bots. Third, to ensure that annotators spend enough time to carefully read the provided texts, we systematically block the ‘submit’ button until 10 minutes have been reached. Lastly, we purposely add three dummy questions consisting of irrelevant text pairs. These questions serve to filter the annotators who do not fully understand the task or submit random responses without reading. We repeat this procedure until all batches have three approved responses.

Each sample from our final annotation consists of three labels rated by multiple annotators. To measure the consistency and reliability of their evaluation, we compute the inter-rater agreement scores by counting the proportion of labels that all annotators agreed to. The scores are as follows: 87.15% for plagiarism-free cases, 51.64% for paraphrase plagiarism cases, and 75.32% for summary plagiarism cases. The relatively low score for paraphrase plagiarism may be explained by the

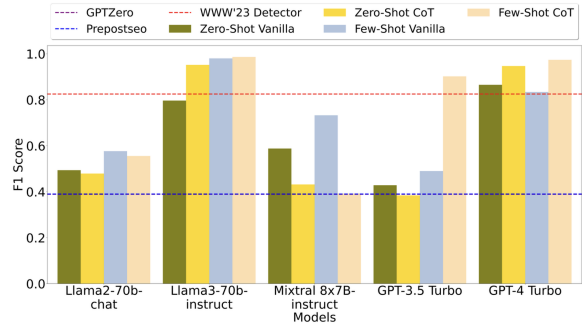


Figure 2: Binary plagiarism detection (no plagiarism vs. plagiarism) performance of 5 LLMs w.r.t. prompt types. Dotted lines represent the performance of non-LLM based detectors.

complexity and subjectivity involved in identifying paraphrased content, as human raters may disagree on how much rewording is acceptable to preserve semantic equivalence. Based on the manual inspection, when the LLM-generated text does not have many word overlaps with the source text with increased complexity, the agreement rate is prone to diverge. To address this misalignment, we filter out those samples with below 50% agreement scores.

**Manual Filtering Results.** After filtering samples with low inter-rater agreement scores, the final data contains 1,239, 545, and 1,181 samples for the plagiarism-free (94.95% agreement rates), paraphrase (88.01% agreement rates), and summary (85.78% agreement rates) plagiarism labels. We take the majority votings of three annotators’ responses and consider them as gold standard labels per evaluation metrics. Consistent with automatic evaluation filtering, we further remove samples that fail to satisfy all 3 or 4 aspects, depending on the plagiarism category. Table 7 illustrates the document percentage breakdown per plagiarism type and domain type after manual filtering.

## 4 PlagBench Plagiarism Detection Task

**Taxonomy of Plagiarism.** we focus on detection of extrinsic plagiarism detection that involves directly comparing a suspicious document  $D_{susp}$  against a source document  $D_{src}$  to detect plagiarism instances (Alzahrani et al., 2011). Following the plagiarism detection task proposed by PAN,<sup>9</sup> we study three branches of plagiarism: *verbatim*, *paraphrase*, and *summary*. Verbatim plagiarism occurs when exact copies of words or phrases from  $D_{src}$  are found in  $D_{susp}$  without quotation marks. Paraphrase plagiarism, on the other hand, is when

<sup>9</sup><https://pan.webis.de>

$D_{susp}$  rephrases  $D_{src}$  using different words but retaining the same meaning and structure without providing a citation. Synonymous substitution, word reordering, and insertion/deletion are common paraphrasing techniques used by plagiarists (Alvi et al., 2021). While both verbatim and paraphrase plagiarism cases tend to maintain the same length and structure as the original text, summary plagiarism involves succinctly summarizing  $D_{src}$  to reuse its key points or ideas. As summaries leave out unnecessary information, they are prone to be significantly shorter than  $D_{src}$ . Unlike verbatim plagiarism, which can be effectively captured using simple string-matching algorithms, paraphrase, and summary plagiarism are more challenging to identify due to their limited lexical and syntactic similarities to the original text. Specifically, these two categories require careful attention to the meaning and context of the content rather than just high-level vocabulary overlaps.

**Task Formulation.** To investigate LLMs’ capabilities in plagiarism detection, we present two tasks: 1) binary classification (plagiarism-free vs. plagiarism) and 2) plagiarism type identification (plagiarism-free vs. verbatim vs. paraphrase vs. summary plagiarism). We hypothesize that the second task is more difficult than the first task because the detector must capture more nuanced differences among the types of plagiarism. Our test set consists of 810 pairs of  $D_{susp}$  and  $D_{src}$  randomly sampled from human annotation results. Specifically, it has 45 text pairs from no plagiarism labels and 45 from three plagiarism labels, generated by three LLMs within three writing domains.

**Experiment Setup.** Using the 810 test set, we evaluate Five LLMs, Llama2-70b-chat, Llama3-70b-instruct, Mixtral-8x7B-instruct, GPT-3.5 Turbo, and GPT-4 Turbo, for binary classification of plagiarism and plagiarism type identification. In addition, three non-LLM-based plagiarism detectors are included for evaluation. Two of these, GPTZero<sup>10</sup> and Prepostseo,<sup>11</sup> are commercial plagiarism checkers, while the third is a detector proposed by Lee et al. (2023b). We denote Lee et al. (2023b)’s detector as the WWW’23 detector for the rest of the paper. All these tools rely on semantic similarity measures and text alignment algorithms to distinguish plagiarism.

For LLM evaluation, we experiment with four

<sup>10</sup><https://gptzero.me/>

<sup>11</sup><https://www.prepostseo.com/plagiarism-checker>

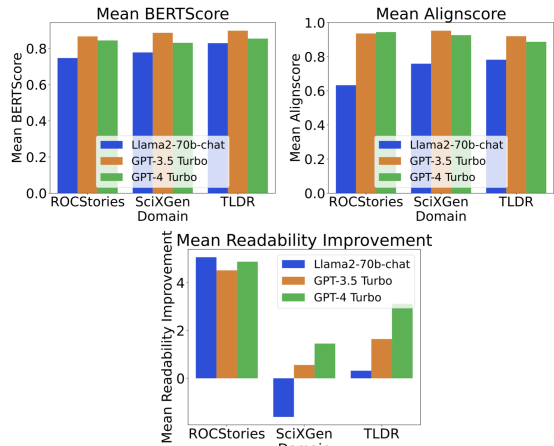


Figure 3: Mean paraphrase evaluation aspect scores w.r.t. domain and model types (automatic evaluation). For BERTScore and Alignscore, a higher score indicates greater semantic similarity and consistency between the LLM-paraphrased text and the source text. A higher readability improvement score suggests that the LLM-paraphrased text is more complex and requires a higher level of education to understand compared to the source text.

popular prompting techniques: zero-shot vanilla prompting, zero-shot Chain-of-Thought (CoT) prompting (Wei et al., 2022), few-shot vanilla prompting, and few-shot CoT prompting (Table 9). CoT prompting is one of the most popular prompt techniques to enhance models’ downstream task performance by eliciting reasoning. As shown in Table 9, we define the definitions of plagiarism categories inside the prompt, aiming to enhance the alignment of LLMs with our task. For few-shot experiments, we provide six demonstrations, a mix of three plagiarism pairs and three plagiarism-free pairs. We leverage GPT-4 to obtain high-quality reasoning for few-shot CoT prompting. For generation, we employ greedy decoding (i.e., choosing the most plausible token at each generation step) to support the reproducibility of our results to some extent. Finally, we report F1 scores for detection performance.

## 5 Experiment Results

### 5.1 RQ1. how well can LLMs generate paraphrase and summary plagiarism?

**Automatic Paraphrase Evaluation.** The purpose of this analysis is to comprehend how accurately Llama2-70b-chat, GPT-3.5 Turbo, and GPT-4 Turbo can perform paraphrasing and summarization. Hence, we examine automated evaluation scores of their generations *prior to* data filtering.

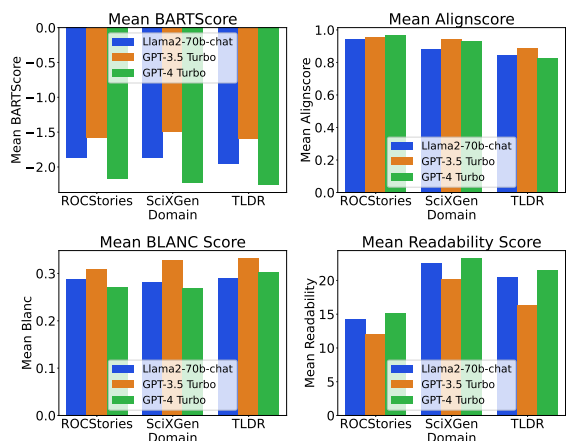


Figure 4: Mean summary evaluation aspect scores w.r.t. domain and model types (automatic evaluation). The lower the BARTScore the more coherent given a pair of the source text and the LLM-summarized text. For Alignscore and BLANC score, a higher score indicates greater consistency and relevancy between the LLM-summarized text and the source text. A higher readability score suggests that the LLM-summarized text is more complex and requires a higher level of education to understand compared to the source text.

Figure 3 shows mean evaluation scores with respect to types of models and writing domains in paraphrase generation. We exclude the grammatical analyses since the majority of LLM-reworded texts are grammatically correct. The results indicate that Llama2-70b-chat achieves significantly lower performance in BERTScore and Alignscore compared to both GPT-3.5 Turbo and GPT-4 Turbo. The observed pattern persists across three writing domains. Particularly, Llama2-70b-chat suffers from generating factually consistent paraphrases for short stories (ROCStories). This finding is consistent with existing literature (Mishra et al., 2024; Dahl et al., 2024) highlighting the vulnerability of Llama2 in hallucination. Within the scope of semantic equivalence and consistency metrics, GPT-3.5 Turbo achieves the highest score, even occasionally better than GPT-4 Turbo. Analyses on readability score<sup>12</sup> gaps between the human-written source text and the machine-generated suspicious text suggest that all three models significantly complicate ROCStories during rephrasing. In regards to SciXGen and TLDR, GPT models tend to produce more sophisticated texts, whereas Llama2-70b-chat often degrades linguistic quality. Substantial readability

<sup>12</sup>We use the automated readability index to measure readability. A higher readability index hints that the text is more complex and may necessitate a higher level of education or reading proficiency to comprehend.

gaps, especially in the GPT-4 version, are somewhat anticipated, as Onder et al. (2024); Momenaei et al. (2023) have noted that ChatGPT’s outputs can be challenging to understand, often demanding a college-level proficiency in linguistic skills.

**Automatic Summary Evaluation.** Summary evaluation results are illustrated in Figure 4. Regardless of writing domain categories, GPT-3.5 Turbo is ranked the lowest in BARTScore, followed by Llama2-70b-chat and then GPT-4 Turbo. The lower the BARTScore the more coherent given a pair of the source document and the machine-summarized document. There are no noticeable differences between the three models regarding the Alignscore, suggesting that they excel at producing factually consistent summaries. Still, GPT-3.5 Turbo outperforms both models in the domains of SciXGen and TLDR. Also, BLANC scores are shown to be the highest for GPT-4 Turbo while GPT-4 Turbo was ranked the lowest. The higher the BLANC score is the more relevant and useful the given summary is. As opposed to paraphrase analyses which take into account the difference between two pieces of text, here we only compute the readability index scores of LLM-generated summaries. This is because we do not have reference summaries to compare against. Consistent with paraphrase generation, GPT-4 Turbo tends to produce summaries with higher readability scores in comparison to Llama2-70b-chat and GPT-3.5 Turbo. Interestingly, summaries generated by Llama2-70b-chat are associated with higher readability scores than GPT-3.5 Turbo. These findings overall suggest that GPT-3.5 Turbo is generally the best paraphraser and summarizer in terms of all aspects, except for fluency/readability. Also, while Llama2-70b-chat is ranked the worst model for paraphrasing tasks, GPT-4 Turbo suffers the most for providing high-quality summaries.

## 5.2 RQ2. how well can LLMs detect plagiarism?

**Binary Plagiarism Detection Results.** Figure 2 shows the plagiarism detection performance of  $D_{src}$  and  $D_{susp}$  in a binary setting (original vs. plagiarism). The results of zero-shot vanilla prompting, the most basic setup, hint that Llama2-70b-chat and GPT-3.5 Turbo perform almost close to a random guess ( $F1 = 0.5$ ). According to our qualitative inspection, the errors from Llama2-70b-chat result from the model not being able to predict a fi-

Detectors	N/A	Llama2-70b-chat			GPT-3.5 Turbo			GPT-4 Turbo			Acc
	Verb	No	Para	Summ	No	Para	Summ	No	Para	Summ	
Llama2-70b-chat	66.67%	36.29%	68.88%	6.66%	30.37%	55.55%	8.88%	33.33%	71.11%	0%	39.5%
Llama3-70b-instruct	<b>100%</b>	85.92%	<b>95.55%</b>	11.11%	81.48%	<b>100%</b>	11.11%	78.51%	<b>100%</b>	8.88%	75.06%
Mixtral 8x7B-instruct	88.14%	34.81%	55.55%	11.11%	29.62%	66.66%	8.88%	21.48%	77.77%	11.11%	41.85%
GPT-3.5 Turbo	77.77%	47.41%	86.66%	<b>17.77%</b>	42.22%	97.77%	2.22%	41.48%	91.11%	2.22%	51.35%
GPT-4 Turbo	99.25%	<b>93.33%</b>	82.22%	<b>17.77%</b>	<b>92.59%</b>	<b>100%</b>	<b>15.55%</b>	<b>85.18%</b>	<b>100%</b>	<b>15.55%</b>	<b>80.12%</b>
GPTZero	37.03%	<b>100%</b>	6.66%	0%	<b>100%</b>	2.22%	4.44%	<b>100%</b>	0%	0%	56.91%
Prepostseo	51.85%	97.03%	20%	2.22%	96.29%	17.77%	2.22%	95.55%	4.44%	6.66%	60.2%
WWW'23 detector	<b>82.22%</b>	80%	<b>88.88%</b>	<b>82.22%</b>	77.77%	<b>86.66%</b>	<b>84.44%</b>	83.7%	<b>77.77%</b>	<b>88.88%</b>	<b>82.34%</b>

Table 2: Plagiarism type classification (original (i.e., ‘No’) vs. verbatim (i.e., ‘Verb’) vs. paraphrase (i.e., ‘Para’) vs. summary (i.e., ‘Summ’) plagiarism) performance of 8 detectors w.r.t. the models used for generation. For non-LLM approaches, we compute the category-wise breakdown from binary classification as they are not suitable for this task. ‘Acc’ in the rightmost column indicates the overall accuracy regardless of category types. The highest values among LLM-based approaches are highlighted in **red**, while the highest values for non-LLM-based approaches are highlighted in **blue**.

nal binary output. 26.7% of Llama2-70b-instruct’s output does not provide the prediction at all. Additionally, its prediction tends to be highly skewed to the negative class. GPT-3.5 Turbo, on the other hand, generally can generate a binary prediction. However, it suffered from low recall (0.3). Comparing across the same families (Llama2 vs. Llama3 and GPT-3.5 vs. GPT-4), there is a significant performance gap. These findings are reasonable in the sense that Llama3 and GPT-4 are upgraded versions of Llama2 and GPT-3.5, respectively. A similar phenomenon has been reported in the task of medical final examination (Rosol et al., 2023). We also find that some models achieve higher performance with CoT prompting and few-shot learning compared to zero-shot vanilla prompting. These results are consistent with previous literature (Wei et al., 2022; Brown et al., 2020). In particular, Llama3-70b-instruct and GPT-4 Turbo achieve near-perfect performance. Yet, the remaining models often obtain incorrect responses due to hallucination. For instance, although their reasoning steps are logically valid, their final response contradicts them.

Commercial plagiarism detectors, GPTZero and Prepostseo, are shown to achieve very low F1 scores. This may be due to two reasons: (1) their built-in retrieval systems fail to fetch relevant documents from the Internet; these tools rely on simple string-matching algorithms designed specifically for detecting verbatim plagiarism, rather than accommodating a broader range of plagiarism categories. Conversely, the WWW’23 detector exhibits stronger performance as it is specifically tailored for this particular task, extrinsic plagiarism detection. Nonetheless, four LLMs have been shown to surpass their performance through few-shot CoT

prompting. Overall, these results highlight the potential of LLMs in effectively detecting plagiarism.

**Plagiarism Type Classification Results.** Table 2 shows detection results of 8 detectors on plagiarism type classification depending on plagiarism types and models used for generation. Of the four prompting techniques, we resort to zero-shot CoT prompting for this experiment. This decision is motivated by the fact that its inference time is significantly faster than few-shot CoT prompting with improved detection performance. Here, due to category-wise performance computation, we report accuracy scores. In line with the results from binary plagiarism detection, Llama3-70b-instruct, and GPT-4 Turbo are the highest-performing LLMs. In particular, their identification of verbatim and paraphrase plagiarism achieved 99-100% accuracy. GPT-3.5 Turbo is demonstrated to be quite capable of distinguishing paraphrase plagiarism, but it performs poorly in identifying plagiarism-free content and summarized content. Most notably, LLMs perform poorly in detecting summary plagiarism across all genres. Most of their errors stem from short texts being confused for paraphrase plagiarism instead of summary plagiarism.

Now, looking at traditional plagiarism checkers, we find that GPTZero and Prepostseo excel at identifying non-plagiarized content compared to the WWW’23 detector. Yet, the WWW’23 detector surpasses them in paraphrase and summary detection. This discrepancy may be due to differing definitions of plagiarism; since these tools are designed for detecting verbatim plagiarism, they may incorrectly classify text pairs with minimal lexical overlap as non-plagiarized.



## 6 Conclusion

We present a novel large-scale plagiarism detection benchmark, **PlagBench**: a collection of 46.5K artificial plagiarism cases generated by three cutting-edge LLMs. This underscores the nuanced challenges and opportunities presented by LLMs in combating plagiarism while also acknowledging their potential vulnerabilities to being used for unethical practices. We envision **PlagBench** as a universal standard for evaluating plagiarism detection methods.

## Limitations

The current research has two limitations. First, we use a single manually crafted prompt template for generating machine-plagiarized documents and detecting plagiarism. There are, however, various prompting techniques and strategies for automatically optimizing prompts. We expect future work to explore a broader range of prompts for both generation and detection purposes. Second, our experiments are based on instruction-tuned decoder-only LLMs. It is uncertain whether the observed findings would apply to other types of LLMs with different architectures. Future research should revisit our RQs using more diverse model types.

## Ethics Statement

This research involves the use of LLMs to simulate plagiaristic behavior. All source documents utilized in this study are derived from publicly accessible open-source datasets. To promote transparency and reproducibility, we will make all data and code used in our experiments available to the research community. We acknowledge the potential ethical concerns associated with generating synthetic plagiarism documents. Therefore, we emphasize that these documents should be used exclusively for research purposes aimed at understanding and mitigating plagiarism. We advise against any misuse of the synthetic data, such as using it to deceive or harm others. Our work is intended to contribute to the development of more effective plagiarism detection tools and to enhance academic integrity. By sharing our resources, we aim to support the broader research community in these endeavors.

Regarding human annotation, our research protocol was approved by the Institutional Review Board (IRB) at our institution. We only recruited annotators that are 18 years old or over. All annotators were paid over minimum wage rate.

## Acknowledgments

This work was in part supported by NSF awards #1934782 and #2114824. Some of the research results were obtained using computational resources provided by CloudBank (<https://www.cloudbank.org/>), which was supported by NSF award #1925001. In addition, this work is partially supported by DHS (17STQAC00001-07-00). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Faisal Alvi, Mark Stevenson, and Paul Clough. 2021. Paraphrase type identification for plagiarism detection using contexts and word embeddings. *International Journal of Educational Technology in Higher Education*, 18(1):42.
- Salha M Alzahrani, Naomie Salim, and Ajith Abraham. 2011. Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):133–149.
- K Avetisyan, G Gritsay, and A Grabovoy. 2023. Cross-lingual plagiarism detection: Two are better than one. *Programming and Computer Software*, 49(4):346–354.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2023. **Quantifying memorization across neural language models**. In *The Eleventh International Conference on Learning Representations*.

- Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.
- David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200.
- Hong Chen, Hiroya Takamura, and Hideki Nakayama. 2021. Scixgen: a scientific paper dataset for context-aware text generation. *arXiv preprint arXiv:2110.10774*.
- Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.
- Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *arXiv preprint arXiv:2401.01301*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.
- Ahmed M Elkhataat, Khaled Elsaid, and Saeed Almeer. 2021. Some students plagiarism tricks, and tips for effective check. *International Journal for Educational Integrity*, 17:1–12.
- Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.
- Tomáš Foltýnek, Terry Ruas, Philipp Scharpf, Norman Meuschke, Moritz Schubotz, William Grosky, and Bela Gipp. 2020. Detecting machine-obfuscated plagiarism. In *International conference on information*, pages 816–827. Springer.
- Bela Gipp and Bela Gipp. 2014. *Citation-based plagiarism detection*. Springer.
- Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.
- Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.
- Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.
- Gunwoo Lee, Jindae Kim, Myung-seok Choi, Rae-Young Jang, and Ryong Lee. 2023a. Review of code similarity and plagiarism detection research studies. *Applied Sciences*, 13(20):11358.
- Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2023b. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, pages 3637–3647.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023. Deepfake text detection in the wild. *arXiv preprint arXiv:2305.13242*.
- Lorelei Lingard. 2023. Writing with chatgpt: An illustration of its capacity, limitations & implications for academic writers. *Perspectives on medical education*, 12(1):261.
- Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

- Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. [Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.
- Frederic P Miller, Agnes F Vandome, and John McBreuster. 2009. Levenshtein distance: Information theory, computer science, string (computer science), string metric, damerau? levenshtein distance, spell checker, hamming distance.
- Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*.
- Bitu Momenaei, Taku Wakabayashi, Abtin Shahlaee, Asad F Durrani, Saagar A Pandit, Kristine Wang, Hana A Mansour, Robert M Abishek, David Xu, Jayanth Sridhar, et al. 2023. Appropriateness and readability of chatgpt-4-generated responses for surgical treatment of retinal diseases. *Ophthalmology Retina*, 7(10):862–868.
- Seyed Vahid Moravvej, Seyed Jalaleddin Mousavirad, Diego Oliva, and Fardin Mohammadi. 2023. A novel plagiarism detection approach combining bert-based word embedding, attention-based lstms and an improved differential evolution algorithm. *arXiv preprint arXiv:2305.02374*.
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.
- 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 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.
- CE Onder, G Koc, P Gokbulut, I Taskaldiran, and SM Kuskonmaz. 2024. Evaluation of the reliability and readability of chatgpt-4 responses regarding hypothyroidism during pregnancy. *Scientific Reports*, 14(1):243.
- Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Martin Potthast, Matthias Hagen, Tim Gollub, Martin Tippmann, Johannes Kiesel, Paolo Rosso, Efsthios Stamatatos, and Benno Stein. 2013. Overview of the 5th international competition on plagiarism detection. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 301–331. CELCT.
- Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. 2010. An evaluation framework for plagiarism detection. In *Coling 2010: Posters*, pages 997–1005.
- Shushanta Pudasaini, Luis Miralles-Pechuán, David Lillis, and Marisa Llorens Salvador. 2024. Survey on plagiarism detection in large language models: The impact of chatgpt and gemini on academic integrity. *arXiv preprint arXiv:2407.13105*.
- Maciej Rosoń, Jakub S Gaşior, Jonasz Łaba, Kacper Korzeniewski, and Marcel Młyńczak. 2023. Evaluation of the performance of gpt-3.5 and gpt-4 on the polish medical final examination. *Scientific Reports*, 13(1):20512.
- Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.
- Timur Sağlam, Moritz Brödel, Larissa Schmid, and Sebastian Hahner. 2024a. Detecting automatic software plagiarism via token sequence normalization. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–13.
- Timur Sağlam, Sebastian Hahner, Larissa Schmid, and Erik Burger. 2024b. Automated detection of ai-obfuscated plagiarism in modeling assignments. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*, pages 297–308.
- RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, Technical report, DTIC document.
- Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation. *arXiv preprint arXiv:2202.08479*.
- Yifan Song, Yuanxin Wang, Marshall An, Christopher Bogart, and Majd Sakr. 2024. Programming plagiarism detection with learner data. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*, pages 1826–1827.
- Benno Stein, Nedim Lipka, and Peter Prettenhofer. 2011. Intrinsic plagiarism analysis. *Language Resources and Evaluation*, 45:63–82.
- Liyan Tang, Zhaoyi Sun, Betina Idnay, Jordan G Nestor, Ali Soroush, Pierre A Elias, Ziyang Xu, Ying Ding, Greg Durrett, Justin F Rousseau, et al. 2023. Evaluating large language models on medical evidence summarization. *npj Digital Medicine*, 6(1):158.

- Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- K Vani and Deepa Gupta. 2017. Detection of idea plagiarism using syntax–semantic concept extractions with genetic algorithm. *Expert Systems with Applications*, 73:11–26.
- Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20, Online. Association for Computational Linguistics.
- Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. How large language models are transforming machine-paraphrase plagiarism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 952–963, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jan Philip Wahle, Terry Ruas, Norman Meuschke, and Bela Gipp. 2021. Are neural language models good plagiarists? a benchmark for neural paraphrase detection. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 226–229.
- Debora Weber-Wulff, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltýnek, Jean Guerrero-Dib, Oluvide Popoola, Petr Šigut, and Lorna Waddington. 2023. Testing of detection tools for ai-generated text. *International Journal for Educational Integrity*, 19(1):26.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220, Hong Kong. Association for Computational Linguistics.
- Jiale Xiong, Jing Yang, Lei Yan, Muhammad Awais, Abdullah Ayub Khan, Roohallah Alizadehsani, and U Rajendra Acharya. 2024. Efficient reinforcement learning-based method for plagiarism detection boosted by a population-based algorithm for pre-training weights. *Expert Systems with Applications*, 238:122088.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.
- Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.
- Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36:39321–39362.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.
- Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.
- Wanzheng Zhu and Suma Bhat. 2020. GRUEN for evaluating linguistic quality of generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 94–108, Online. Association for Computational Linguistics.

## A Appendix

### A.1 Automatic Evaluation of LLM-Generated Plagiarism

We leverage publicly available two datasets, PAWS (Zhang et al., 2019b) and SummEval (Fabbri et al., 2021), to find the optimal thresholds for data filtering regarding paraphrase evaluation and summary evaluation, respectively. Both datasets contain human annotation labels signaling whether a pair of texts is correct paraphrases or not. PAWS provide one binary label for each text pair, whereas SummEval contains a list of 8 annotation results in which the scores range from 1 to 5 evaluated across 4 dimensions: coherence, consistency, fluency, and relevance.

The threshold selection process is straightforward; we apply our established automated evaluation measurements to the subset of these two

corpora and compute their mean scores of *bad* examples. Specifically, we compute mean scores of entries with ‘0’ label from the PAWS dataset and entries with average annotation scores lower than 2.5 from the SummEval dataset. These mean scores are then used as a reference to assign filtering thresholds. Given two pieces of evaluation texts  $(d, d^*)$ , let’s denote paraphrase evaluation results as  $R_{\text{para\_eval}}$ .  $R_{\text{para\_eval}}$  can be expressed as below:

$$R_{\text{para\_eval}} = \begin{cases} 1 & \text{if Alignscore}(d, d^*) > 0.5 \\ & \wedge \text{BERTScore}(d, d^*) > 0.5 \\ & \wedge \text{Readability\_Gap}(d, d^*) > -0.5 \\ & \wedge \text{COLAScore\_Improv}(d, d^*) \geq 0 \\ 0 & \text{else.} \end{cases}$$

Summary evaluation results are denoted as  $R_{\text{summ\_eval}}$ .

$$R_{\text{summ\_eval}} = \begin{cases} 1 & \text{if Alignscore}(d, d^*) > 0.5 \\ & \wedge \text{BLANC}(d, d^*) > 0.0 \\ & \wedge \text{BARTScore}(d, d^*) > -4.0 \\ & \wedge \text{Readability}(d, d^*) > 10 \\ & \wedge \text{COLAScore\_Improv}(d, d^*) \geq 0 \\ 0 & \text{else.} \end{cases}$$

Given these two definitions, we omit all samples associated with ‘0’ in regards to  $R_{\text{para\_eval}}$  and  $R_{\text{summ\_eval}}$ .

## A.2 Details on Human Evaluation of LLM-Generated Plagiarism

**Task Formation.** Each task consists of three batches of one source text and three suspicious texts generated by Llama2-70b-chat, GPT-3.5 Turbo, and GPT-4 Turbo. Given a pair of the source text and machine-paraphrased or machine-summarized texts, we first verbally describe our established evaluation aspects on the task description and ask them to answer if the described aspect is satisfied or not. For plagiarism-free cases, the definitions of paraphrase and summary plagiarism are provided, and annotators are tasked to identify two texts belong to either paraphrase or summary plagiarism. See Table 3, and 4, and 5. Given that the estimated completion time per batch is 10 minutes, workers are compensated \$1.5 per batch based on United States average hourly wages.

## A.3 Model Size And Budget

We enlist model size used for our experiment here: for Llama2 and Llama3, we use the largest scale among their models, which is 70B. GPT-3.5 is expected to contain 175B parameters, equivalent to

GPT-3. Meanwhile, GPT-4 has 1.76T parameters, and Mixtral has 46.7B total parameters. For all experiments, we leverage API calls from Huggingface and OpenAI official website.

---

**Instruction**

---

In this HIT, you will be presented with 5 paragraphs consisting of 1 source text and 4 machine-generated texts covering a similar topic as the source text. In total, there will be 3 sets of 5 paragraphs for you to evaluate. Please read the texts carefully and answer the following questions to evaluate if the provided texts plagiarize the source text. There are 2 plagiarism types you need to consider:

- **Paraphrase plagiarism:** The evaluation text is rephrased or rewritten using different words but retains the same meaning and structure as the source text.
  - **Summary plagiarism:** The evaluation text encapsulates the most essential points of the source text into a shorter form.
- 

**FIRST SET**

Source Text: \${source\_doc}

Evaluation Text A: \${evaluation\_doc}

---

**Questions for Evaluation Text A**

**Q1. Paraphrase plagiarism:** Is the evaluation text a paraphrase of the source text (i.e., rephrased or rewritten using different words but retaining the same meaning and structure)?

**Answer:** Yes/No

**Q2. Summary plagiarism:** Does the evaluation text summarize the source text (i.e., encapsulate the most essential points of the source text into a shorter form)?

**Answer:** Yes/No

---

Table 3: Human annotation instruction for plagiarism-free cases

---

**Instruction**

---

In this HIT, you will be presented with 5 paragraphs consisting of 1 source text and 4 corresponding machine-generated texts. In total, there will be 3 sets of 5 paragraphs for you to evaluate. Please read the texts carefully and answer the following questions to evaluate the overall quality of the provided texts given the source text.

---

**FIRST SET**

Source Text: \${source\_doc}

Evaluation Text A: \${evaluation\_doc}

---

**Questions for Evaluation Text A**

**Q1. Relevance:** Does the evaluation text cover all the essential information and key points (without including irrelevant or tangential details) from the source text?

**Answer:** Yes/No

**Q2. Consistency:** Does the evaluation text accurately represent the facts, details, and information contained in the source text?

**Answer:** Yes/No

**Q3. Coherence:** Is the evaluation text well-structured and not just a random assortment of information from the source text?

**Answer:** Yes/No

---

Table 4: Human annotation instruction for paraphrase plagiarism cases

---

**Instruction**

---

In this HIT, you will be presented with 5 paragraphs consisting of 1 source text and 4 corresponding machine-generated texts. In total, there will be 3 sets of 5 paragraphs for you to evaluate. Please read the texts carefully and answer the following questions to evaluate the overall quality of the provided texts given the source text.

---

**FIRST SET**

Source Text: \${source\_doc}

Evaluation Text A: \${evaluation\_doc}

---

**Questions for Evaluation Text A**

**Q1. Relevance:** Does the evaluation text cover all the essential information and key points (without including irrelevant or tangential details) from the source text?

**Answer:** Yes/No

**Q2. Consistency:** Does the evaluation text accurately represent the facts, details, and information contained in the source text?

**Answer:** Yes/No

**Q3. Coherence:** Is the evaluation text well-structured and not just a random assortment of information from the source text?

**Answer:** Yes/No

**Q4. Language quality:** Does the evaluation text maintain or improve upon the fluency and grammaticality of the source text?

**Answer:** Yes/No

---

Table 5: Human annotation instruction for summary plagiarism cases

Domain	Model	# of document pairs	# of remaining pairs (paraphrase)	# of remaining pairs (summary)
SciXGen	Llama2-70b-chat	2,400	1,179 (49.12%)	2,147 (89.46%)
	GPT-3.5 Turbo	2,400	1,764 (73.5%)	2,379 (99.12%)
	GPT-4 Turbo	1,300	1,085 (83.46%)	1,266 (97.38%)
ROCStories	Llama2-70b-chat	2,400	1,699 (70.79%)	1,975 (82.29%)
	GPT-3.5 Turbo	2,400	2,385 (99.38%)	1,614 (67.25%)
	GPT-4 Turbo	1,300	1,295 (99.62%)	1,166 (89.69%)
TLDR	Llama2-70b-chat	1300	753 (57.92%)	1,060 (81.54%)
	GPT-3.5 Turbo	1300	1,230 (99.38%)	1,181 (90.85%)
	GPT-4 Turbo	700	681 (97.29%)	657 (93.86%)

Table 6: Dataset statistics after automatic filtering for paraphrase and summary plagiarism. The percentage in the bracket represents the percentage of remaining document pairs remaining out of original pairs.

Domain	Model	# of document (plagiarism-free)	# of document (paraphrase)	# of document (summary)
SciXGen	Llama2-70b-chat	108	83	102
	GPT-3.5 Turbo	99	87	132
	GPT-4 Turbo	118	45	129
ROCStories	Llama2-70b-chat	196	55	118
	GPT-3.5 Turbo	196	45	120
	GPT-4 Turbo	197	56	117
TLDR	Llama2-70b-chat	114	55	97
	GPT-3.5 Turbo	107	56	114
	GPT-4 Turbo	113	45	129

Table 7: Dataset statistics after manual filtering for plagiarism labels.

Plagiarism category	Template
Paraphrase	Paraphrase the following text while keeping its meaning. Text: $\{\text{source\_doc}\}$ Paraphrased:
Summary	Summarize the following text in 1-3 sentences. Text: $\{\text{source\_doc}\}$ Summarized:
Plagiarism-Free	Based on the provided text passage and keywords, write its continuation in a style of $\{\text{genre}\}$ . When generating, make sure that the continuation is coherent while including all keywords. Text: $\{\text{start\_sent}\}$ Keywords: $\{\text{keyword}\}$ Continuation:

Table 8: Prompt templates for plagiarism generation task.

Prompt type	Template
<b>Zero-shot vanilla prompting (binary detection)</b>	<p>There are three types of plagiarism:</p> <ul style="list-style-type: none"> <li>• Verbatim plagiarism: the evaluation text consists of exact copies of words or phrases without transformation from the source text without citation.</li> <li>• Paraphrase plagiarism: the evaluation text is rephrased or rewritten using different words but retain the same meaning and structure as the source text without citation.</li> <li>• Summary plagiarism: the evaluation text encapsulates the most essential points of the source text into a shorter form without citation.</li> </ul> <p>If the evaluation text does not belong to any of three plagiarism categories, it means plagiarism-free. Given a pair of text and provided plagiarism definitions, does the evaluation text plagiarize the source text (yes/no)? Please format your final response as 'Answer: {{response}}'.</p> <p>Source text: \${source_doc} Evaluation text: \${evaluation_doc}</p>
<b>Zero-shot CoT prompting (binary detection)</b>	<p>There are three types of plagiarism:</p> <ul style="list-style-type: none"> <li>• Verbatim plagiarism: the evaluation text consists of exact copies of words or phrases without transformation from the source text without citation.</li> <li>• Paraphrase plagiarism: the evaluation text is rephrased or rewritten using different words but retain the same meaning and structure as the source text without citation.</li> <li>• Summary plagiarism: the evaluation text encapsulates the most essential points of the source text into a shorter form without citation.</li> </ul> <p>If the evaluation text does not belong to any of three plagiarism categories, it means plagiarism-free. Given a pair of text and provided plagiarism definitions, does the evaluation text plagiarize the source text (yes/no)? First think step-by-step and format your final response as 'Final answer: {{response}}'.</p> <p>Source text: \${source_doc} Evaluation text: \${evaluation_doc}</p> <p>Answer: Let's think step-by-step.</p>
<b>Zero-shot vanilla prompting (type identification)</b>	<p>There are three types of plagiarism:</p> <ul style="list-style-type: none"> <li>• Verbatim plagiarism: the evaluation text consists of exact copies of words or phrases without transformation from the source text without citation.</li> <li>• Paraphrase plagiarism: the evaluation text is rephrased or rewritten using different words but retain the same meaning and structure as the source text without citation.</li> <li>• Summary plagiarism: the evaluation text encapsulates the most essential points of the source text into a shorter form without citation.</li> <li>• No plagiarism: the evaluation text does not belong to any of three plagiarism categories.</li> </ul> <p>Given a pair of text and provided plagiarism definitions, what type of plagiarism (no/verbatim/paraphrase/summary) does the evaluation belong to when compared to the source text? Please format your final response as 'Answer: {{response}}'.</p> <p>Source text: \${source_doc} Evaluation text: \${evaluation_doc}</p>
<b>Zero-shot CoT prompting (type identification)</b>	<p>There are three types of plagiarism:</p> <ul style="list-style-type: none"> <li>• Verbatim plagiarism: the evaluation text consists of exact copies of words or phrases without transformation from the source text without citation.</li> <li>• Paraphrase plagiarism: the evaluation text is rephrased or rewritten using different words but retain the same meaning and structure as the source text without citation.</li> <li>• Summary plagiarism: the evaluation text encapsulates the most essential points of the source text into a shorter form without citation.</li> <li>• No plagiarism: the evaluation text does not belong to any of three plagiarism categories.</li> </ul> <p>Given a pair of text and provided plagiarism definitions, what type of plagiarism (no/verbatim/paraphrase/summary) does the evaluation belong to when compared to the source text? First think step-by-step and format your final response as 'Answer: {{response}}'.</p> <p>Source text: \${source_doc} Evaluation text: \${evaluation_doc}</p> <p>Answer: Let's think step-by-step.</p>

Table 9: Prompt templates for plagiarism detection task.