

Understanding the effects of human-written paraphrases in LLM-generated text detection

Hiu Ting Lau, Arkaitz Zubiaga*

School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, United Kingdom



ARTICLE INFO

Keywords:

LLM-generated text detection
Human-written paraphrases
Large language models

ABSTRACT

Natural Language Generation has been rapidly developing with the advent of large language models (LLMs). While their usage has sparked significant attention from the general public, it is important for readers to be aware when a piece of text is LLM-generated. This has brought about the need for building models that enable automated LLM-generated text detection, with the aim of mitigating potential negative outcomes of such content. Existing LLM-generated detectors show competitive performances in telling apart LLM-generated and human-written text, but this performance is likely to deteriorate when paraphrased texts are considered. In this study, we devise a new data collection strategy to collect Human & LLM Paraphrase Collection (HLPC), a first-of-its-kind dataset that incorporates human-written texts and paraphrases, as well as LLM-generated texts and paraphrases. With the aim of understanding the effects of human-written paraphrases on the performance of SOTA LLM-generated text detectors OpenAI RoBERTa and watermark detectors, we perform classification experiments that incorporate human-written paraphrases, watermarked and non-watermarked LLM-generated documents from GPT and OPT, and LLM-generated paraphrases from DIPPER and BART. The results show that the inclusion of human-written paraphrases has a significant impact of LLM-generated detector performance, promoting TPR@1%FPR with a possible trade-off of AUROC and accuracy.

1. Introduction

Large language models (LLMs) have become essential in Natural Language Processing (NLP) thanks to their advanced capabilities for text processing and generation, which is achieved through analysis of patterns and relationships between words and sentences using transformer models (Zubiaga, 2024). Consequently, LLMs have had a significant impact on Natural Language Generation (NLG), as they have provided improved capacity for automatically generating high quality text (Barreto et al., 2023).

While the advancement of LLMs in the context of NLG has aided tasks such as machine translation (Zhang et al., 2023) and text summarization (Zhang et al., 2024), it has also given rise to undesired social problems, including intentional malicious usage, ethical concerns and information inaccuracy. This has brought about the need for researching the development of methods for automated LLM-generated text detection which distinguishes if a text is human- or LLM-generated (Fröhling and Zubiaga, 2021). Currently, there are 2 major streams of LLM-generated text detectors: (i) zero-shot classifiers (Mitchell et al., 2023; Solaiman et al., 2019), which identify LLM-generated text based on the pattern and characteristics of the input, and (ii) watermark detectors (Kirchenbauer et al., 2023), which rely on detecting the

presence of watermarks which are imprinted into the text during the generation process (Kirchenbauer et al., 2023), and are effective in the cases where the watermarks have been added by the LLM. The detectors then examine the input, classifying it as LLM-generated if the level of watermarking exceeds a set threshold, or as human-generated otherwise.

Both kinds of detectors have demonstrated excellent performance in LLM-generated text detection. However, research testing these detectors has primarily focused on datasets involving texts which are exclusively generated by humans or by LLMs. There can be, however, more complicated cases, such as paraphrased texts, which have been seldom considered in previous research. This is important as many LLMs incorporate watermarks in the generated outputs. Watermarking of LLM-generated texts is done by promoting words from the LLM's "green list" and denoting words from the LLM's "red list". Hence, a text which contains more words from the "green list" and fewer words from the "red list" is much more likely to be LLM-generated and hence facilitates detection. Paraphrasing is defined as the rewriting of context in a simpler and shorter form Cambridge Dictionary (2019); an LLM-generated text paraphrased by humans can then lead to the replacement of those "green-list" words by other words, hence

* Corresponding author.

E-mail addresses: ml21305@qmul.ac.uk (H.T. Lau), a.zubiaga@qmul.ac.uk (A. Zubiaga).

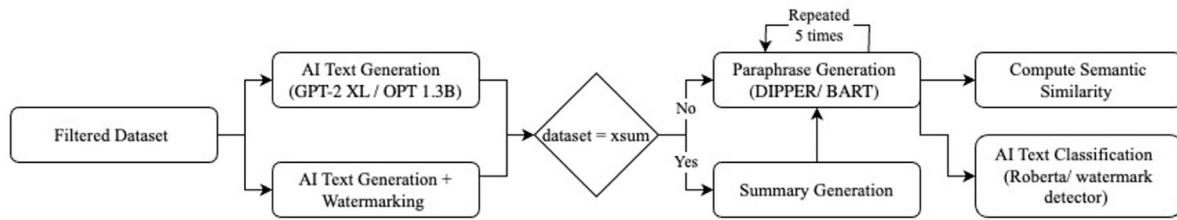


Fig. 1. Flowchart of the LLM text generation, paraphrasing and classification process.

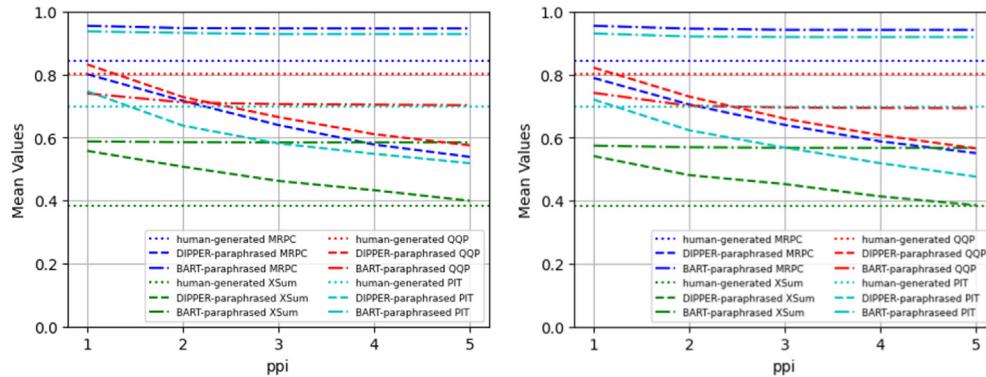


Fig. 2. Mean similarity scores of watermarked (left) and non-watermarked (right) LLM-generated texts and their paraphrases across different datasets.

changing its statistical properties and reducing its detectability. Since the above detectors perform classification based on token patterns and watermarks, paraphrasing could effectively evade both zero-shot classifiers and watermark detectors while preserving semantic information from the original LLM-generated text. It is important to identify that a text originated from an LLM, despite having been subsequently paraphrased, as this can still be leveraged for massive generation of texts for malicious purposes.

In this work, we are the first to comprehensively study the effectiveness of LLM-generated text detectors in the presence of human-paraphrased texts, in turn assessing the impact of these edited texts on the model performance. In this study, we aim to address this problem by using human-written paraphrases for classification, with the notion that human-written paraphrases and LLM-generated paraphrases might contain different characteristics, which potentially improve the classifiers' performances. We set forth the following research question: “What are the effects of including human-written paraphrases in LLM-generated text detection?” With this aim and research question in mind, we make the following contributions:

- We perform a review of the literature to investigate the existing NLG developments, the importance of LLM-generated text detection, and the performances of existing detectors.
- We describe a data collection process which enables us to build and release the Human & LLM Paraphrase Collection (HLPC) dataset with human-generated and LLM-generated documents, along with their paraphrases.
- We perform classification experiments using SOTA AI paraphrasers and detectors, along with human-written paraphrases.
- Our study contributes to the domain of LLM-generated text detection by examining the effects of including human-written paraphrases in classification and providing insights on data inclusion in future detector building.

2. Related work

We review existing research in NLG with a focus on LLMs, the importance of LLM-generated text detection and existing detection models, following with a discussion of the main research gaps that our study addresses.

2.1. Natural Language Generation (NLG)

NLG represents a substantial branch of research within NLP, where existing NLG tasks include question-answering (Allam and Haggag, 2012), text summarization (El-Kassas et al., 2021), and machine translation (Lopez, 2008).

LLMs in NLG. Very recently, the development of LLMs has brought significant improvements to NLG, primarily due to their ability of learning linguistic patterns from very large-scale corpora. Before the adoption of LLMs, two techniques were used for NLG. The earliest NLG systems used templates and rules (Mou, 2022) and the later systems utilized conditional probability between words to account for context dependency (Goyal et al., 2023). Fundamentally, these systems lack flexibility and adaptability since text generation is restricted, resulting in unfavorable generation patterns, such as inaccuracy in question answering in the rule-based model and word repetitions in the probabilistic model.

Recent research proposed different LLMs that adopt deep learning and neural networks in NLP, contributing to a remarkable improvement in NLG (Goyal et al., 2023). With the use of a transformer architecture, the models can capture long-range text dependencies with positional encoding, allowing the models to understand both in-words and in-sentence relationships. The models can also be fine-tuned to cater to specific NLG needs. Coupled with large-scale datasets and parameters, LLMs can understand complex linguistic patterns and relationships. For example, GPT-3 is trained with 499 billion tokens and 175 billion parameters (Brown et al., 2020; Zhang et al., 2022). This promotes

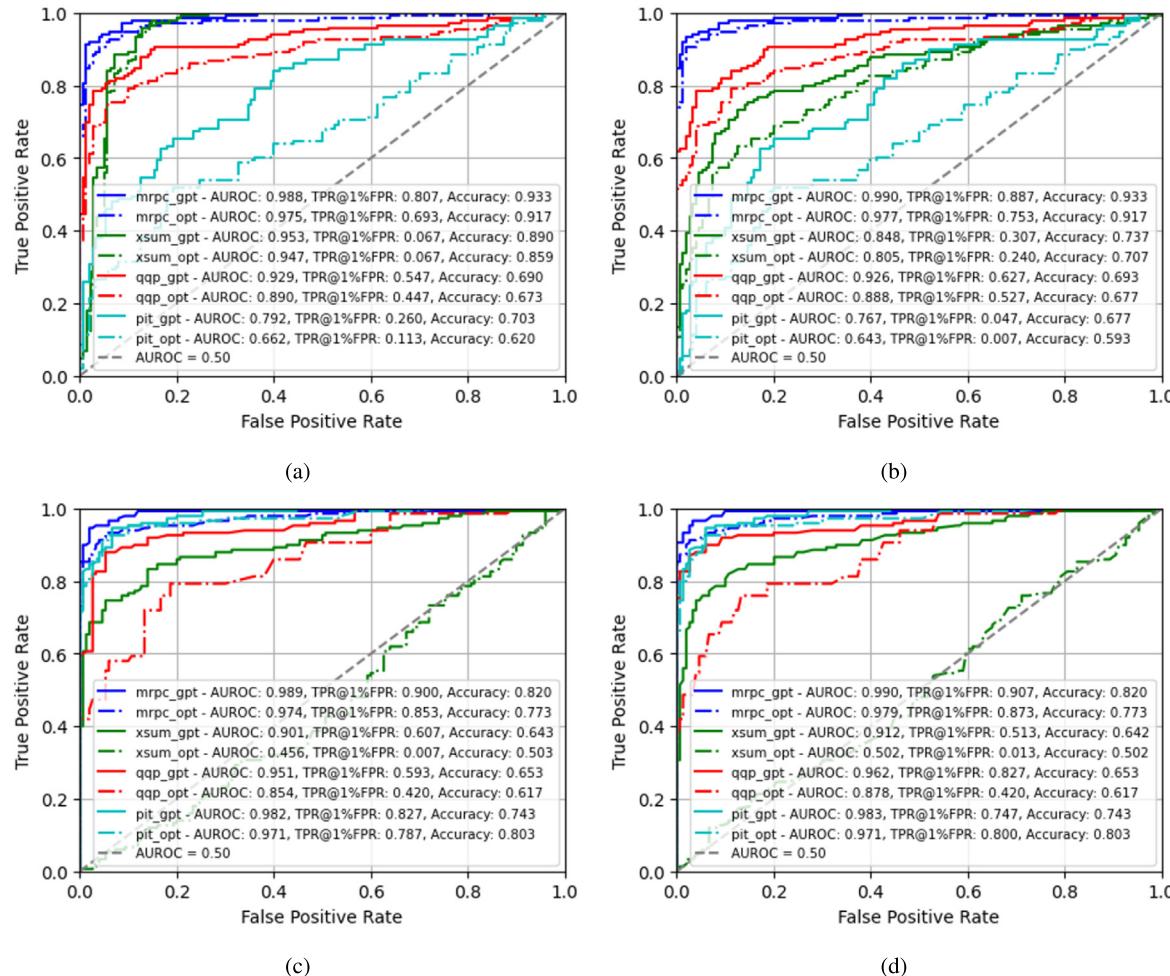


Fig. 3. ROC Curve Comparison: (a) Human-generated Documents vs Non-watermarked LLM-generated Documents, (b) Human-generated Paraphrases vs Non-watermarked LLM-generated Documents, (c) Human-generated Documents vs Watermarked LLM-generated Documents, (d) Human-generated Paraphrases vs Watermark LLM-generated Documents.

models' learning of language representations, including syntactic, contextual and semantic information (Goyal et al., 2023). Text data from large corpora are used to train LLMs, such as CommonCrawl, WebText2, BookCorpus, etc (Brown et al., 2020; Zhang et al., 2022), which allow the models to learn knowledge from a broad range of disciplines. With the above advancements, LLMs can thus generate human-like outputs solely using user prompts as inputs (Krishna et al., 2024). As a result, the use of LLMs along with a simple chat interface has attracted massive usage and attention from the general public (Touvron et al., 2023), with over 180 million users for OpenAI ChatGPT (Shewale, 2023).

While LLMs provide a new direction for text generation in NLG, they are also being widely used to support evaluation of NLG outputs (Li et al., 2024), which is however beyond the scope of this study.

Paraphrase generation. One of the NLG tasks for which LLMs have brought a significant boost is paraphrase generation, with various AI paraphrasers built on top of existing LLMs. Paraphrasing is defined as the rewriting of context in a simpler and shorter form (Cambridge Dictionary, 2019), and is used extensively to avoid content to be flagged as plagiarism. A paraphrase generator takes sentences or paragraphs as inputs, and creates rewritten outputs which preserve the semantics of the original text (Goyal et al., 2023). Currently, there are 2 types of AI paraphrasers: (i) systems that are inherently built for paraphrasing, specifically built to paraphrase text automatically and

evade LLM-generated text detectors; an example of this is DIPPER (Krishna et al., 2024), which could paraphrase long paragraphs and control output diversity, and (ii) AI chatbots that receive paraphrasing prompts to produce paraphrases; an example of this is T5-paraphraser Parrot (Damodaran, 2021).

To account for the quality of paraphrases, grammar accuracy and content semantic preservation are considered either with human experts or automatic model evaluation. In Sadasivan et al. (2023), human evaluation was conducted for the paraphrases generated from DIPPER and Llama-2-7B-Chat. Paraphrases from both models exhibit high ratings in terms of content preservation and grammar accuracy. To account for semantic similarity, Krishna et al. (2024) use the P-SP model to compute semantic similarity scores which reflects the level of contextual relationship between the paraphrases and original text. Jayawardena et al. (2024) use a variety of semantic similarity scores by calculating co-sine distances of the outputs from different embedding models, such as "Ada Score" with embeddings generated from OpenAI's text-embedding-ada-002 (OpenAI, 2023) and "SimCSE Score" with embeddings generated from SimCSE's sup-simcse-roberta-large model (Gao et al., 2021). DIPPER effectively paraphrases text with high semantic similarity. This shows that existing AI paraphrasers, coupled with the use of LLM, have shown great advancement in aiding automatic paraphrase generation.

2.2. Importance of LLM-generated text detection

Despite the convenience brought by LLMs for improved NLG, they can also cause various problems with a negative impact on society, which has motivated the need for investigating methods for LLM-generated text detection. Next, we discuss three key problems arising from the use of LLMs, especially when the generated texts is not flagged or labeled as being LLM-generated: intentional malicious usage, ethical concerns and information inaccuracy (Zhao et al., 2024).

Intentional malicious usage. Malicious actors may exploit LLMs to produce fake content or to produce content in circumstances where LLM-generated content is unacceptable. Examples include the creation of fake news in political elections to denigrate competitors (Illia et al., 2023) and the creation of writing or code to then claim full credit for it in an academic environment (Kirchenbauer et al., 2023). As proposed in Solaiman et al. (2019), malicious actors can be categorized into 3 levels: low-skilled, moderate-skilled and advanced-skilled based on their programming level and available resources. Moderate-skilled actors could already produce fake news or build spambots for social media, let alone the adverse effects that advanced-skilled actors could bring.

Ethical concerns. Ethical considerations concerning gender, race and ethnicity bias are raised attributing to the inherent sampling bias of the LLM, which potentially leads to social inequality and discrimination (An et al., 2024). As presented in Section 2.1, LLMs are trained from very large corpora. However, the sampling of content from these corpora might itself be biased. For example, a study found that Reddit data used to train GPT-2 (Radford et al., 2023) is composed of context generated mainly by young males in the United States (Pew Research Center's Journalism Project, 2016). Since the training of models gives equal weighting for each document in the sample (Illia et al., 2023), the resulting model thus shows an under-representation of female groups and groups in other age ranges. These biases can be further extended to race, ethnicity and disability (Hutchinson et al., 2020; Bender et al., 2021). As a result, LLMs might generate biased content that instills negative stereotypes and sentiments toward certain demographics (Hutchinson et al., 2020), leading to exacerbated social inequality and discrimination.

Information inaccuracy. Information accuracy is not guaranteed by LLMs, which can end up misleading users who are inexperienced or who otherwise overly trust LLM outputs. LLMs can in fact be negatively impacted by inaccurate information that is inherent in the large corpora used for training them. For example, Reddit as a source of GPT-2 (Radford et al., 2023) has a risk of containing information that is not verified, leading to potential information inaccuracy and credibility issues. As a result, LLM-generated text might be inaccurate, even if the information is factual (Mitchell et al., 2023; Illia et al., 2023; Lin et al., 2022). With LLM-generated content being perceived as a credible source by many users (Kreps et al., 2022), it might mislead non-professional users with its outputs of varying levels of quality. Meanwhile, if the problem persists, it will lead to a degradation in LLM-generated text accuracy or cause complications in future training of LLMs.

With the massive creation of LLM-generated content on the web, there is a risk that future training of LLMs could include LLM-generated content without necessarily knowing, which can then amplify accuracy issues on the LLMs. Research has suggested that LLM-generated content should be excluded from training to avoid this problem (Radford et al., 2023) indicating that unnecessary overhead is caused due to the inaccuracy of LLMs. To conclude, considering the increasing usage and potential problems of automatic text generation, it is important to identify LLM-generated text to safeguard the quality of output and to mitigate the aforementioned problems with human oversight.

2.3. SOTA LLM-generated text detection

Approaches to LLM-generated text detection. Currently, there are two major streams of LLM-generated text detectors: (i) zero-shot classifiers (Solaiman et al., 2019; Mitchell et al., 2023), and (ii) watermark detectors (Kirchenbauer et al., 2023). Zero-shot classifiers aim to identify patterns and statistical characteristics of input text, comparing them to those of LLM-generated text (Sadasivan et al., 2023; Wolff and Wolff, 2020). For example, DetectGPT calculates the average log probability ratio of the input text over its perturbations and classifies text as LLM-generated if the ratio exceeds a threshold (Mitchell et al., 2023). Zero-shot classifiers do not require further training, thus facilitating usage from non-technical users. Watermark detectors rely on the addition of a watermark, which is not visible to humans, on LLM-generated text. Kirchenbauer et al. (2023) proposed a watermarking scheme with tokens in “green list” and “red list”. While adding the watermark on LLM-generated text, the use of “green list” tokens are prioritized in the sentence generation process, resulting in a text dominated by these tokens. The detector thus classifies a text to be LLM-generated if the number of “green list” tokens is high. Strong and weak watermarking can be implemented by adjusting the parameters (Kirchenbauer et al., 2023). These two types of detectors both exhibit remarkable performance in LLM-generated text detection.

For zero-shot classifiers, 95% accuracy is observed from RoBERTA fine-tuned classifiers (Solaiman et al., 2019) and 92% from GROVER (Zellers et al., 2019). Apart from accuracy, AUROC is also considered to account for the true positive rate (TPR) and false positive rate (FPR), as false positive is highly discouraged for these classifiers to misclassify human-written text as LLM-generated (Kirchenbauer et al., 2023). DetectGPT shows SOTA performance with an average AUROC of 95.3% on various datasets. It should also be noted that detection performance varied with text length and decoding strategies (Solaiman et al., 2019; Mitchell et al., 2023). Meanwhile, watermark detectors also show outstanding performance, with strong watermarking detection achieving 100% AUROC and soft watermarking achieving 98.9% (Kirchenbauer et al., 2023). Watermark detectors are also regarded as more effective detectors than zero-shot classifiers (Krishna et al., 2024). Other than the above two mainstream LLM-generated text detectors, Krishna et al. (2024) proposed an information retrieval-based detector, which classifies text by comparing the input text with stored LLM outputs.

Paraphrase attacks in LLM-generated text detection. While existing LLM-generated text detectors show remarkable performance, they can be vulnerable to paraphrasing attacks, as one can alter generated texts to circumvent detection. Since these detectors perform classification based on the existence of token patterns or watermarks, paraphrasing on LLM-generated text, either by human or AI paraphrasers, can potentially evade the detectors (Kirchenbauer et al., 2023; Sadasivan et al., 2023; Wolff and Wolff, 2020). Past research has conducted experiments on different types of paraphrasing attacks. Kirchenbauer et al. (2023) performed the attack by replacing words with tokens generated from T5 model and noticed a significant watermark degradation with AUROC decreasing from 99.8% to 69.6%. Krishna et al. (2024) showed that after using DIPPER AI-paraphraser, DetectGPT’s detection rate significantly reduced from 70.3% to 4.6% and watermark detection accuracy decreased from 100% to 57.2%. The aforementioned research performs only a single round of paraphrasing, and it is sufficient to substantially reduce the detectors’ performance. Noticing this, Sadasivan et al. (2023) performed recursive paraphrasing attacks on various LLM-generated text detectors, where LLM-generated text is paraphrased with AI paraphrasers in multiple iterations. They concluded that recursive attacks could further evade these detectors. For non-watermarked text, DetectGPT’s AUROC score decreases from 96.5% to 59.8% with text paraphrased with T5 model. The same outcome also resulted in watermarked text. With 2 rounds of paraphrasing with DIPPER and Llama-2-7B-Chat (Sadasivan et al., 2023),

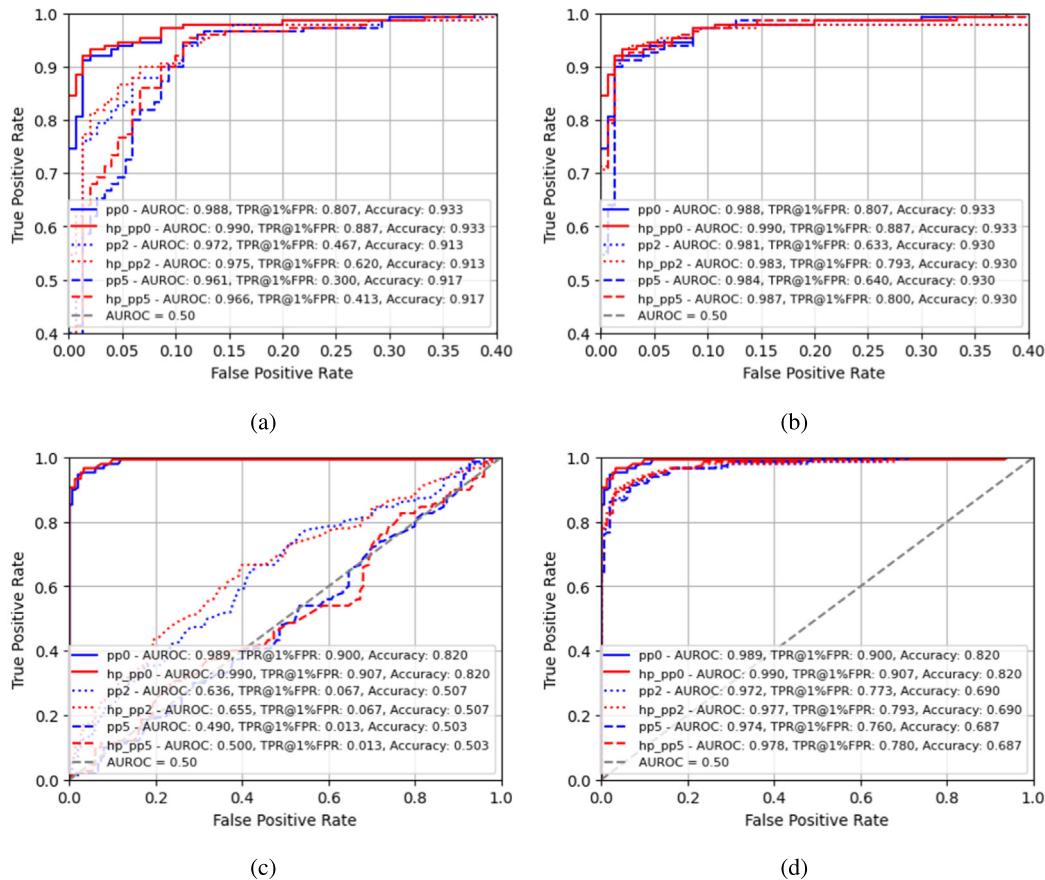


Fig. 4. ROC Curve Comparison: (a) Human-generated Documents/Paraphrases vs DIPPER-generated Paraphrases from Non-watermarked MRPC GPT2-generated Text, (b) Human-generated Documents/Paraphrases vs BART-generated Paraphrases from Non-watermarked MRPC GPT2-generated Text, (c) Human-generated Documents/Paraphrases vs DIPPER-generated Paraphrases from Watermarked MRPC GPT2-generated Text, (d) Human-generated Documents/Paraphrases vs BART-generated Paraphrases from Watermarked MRPC GPT2-generated Text.

TPR@1%FPR drops significantly from 99.8% to 44.8% and 38.9% respectively. The score further decreased to 15.7% and 27.2% respectively after 5 rounds of paraphrasing, reflecting the detectors' inability to identify LLM-generated paraphrases. Other than the degradation in TPR@1%FPR, AUROC scores also decreased from 99.9% to 76.3% and 79.5% respectively. As such, from existing research it can be expected that paraphrasing attacks, which change the statistical properties and replace watermarked tokens, could effectively evade both zero-shot classifiers and watermark detectors while preserving semantic information from the original LLM-generated text (Krishna et al., 2024).

2.4. Research gap

In existing research, classifications are conducted with datasets consisting of human-written text and paraphrases of LLM-generated text. While LLM-generated text detectors make classification based on the existence of LLM-generated features in the input (Sadasivan et al., 2023; Wolff and Wolff, 2020), a potential reason for paraphrasing attack successfully evading these detectors might be the fundamental difference in the language format between LLM-generated text and its paraphrases. Razaq et al. (2024) stated that paraphrases, which contain similar semantic information, could exhibit different lexical, syntactic and word order from the original text. As such, the LLM-generated paraphrases might not contain the characteristics that these detectors are looking for, leading to misclassification.

Hence, to addressing the existing gap, our aim in this research is to include human-paraphrased text in the dataset for classification, which we experiment with SOTA LLM-generated text detectors.

LLM-paraphrased text might potentially distinguish itself from human-paraphrased text, in terms of the degree of semantic understanding, writing style and word choices. As a result, an investigation into the effect of including human paraphrases on LLM-generated text detection can help identify potential reasons for misclassification by existing detectors and to further aid research in developing LLM-generated text detectors.

3. Dataset

3.1. Creating the HLPC dataset

Despite the existence of numerous datasets containing human- and LLM-generated data, we found no suitable dataset that incorporates also their paraphrases which would enable our study, and hence we leverage and extend existing datasets for creating the HLPC dataset.¹

Overview. We put together a dataset consisting of two types of data, each consisting of an original document (DOC) and its paraphrase (PP):

1. Human-generated data: collected from 4 existing datasets, which include MRPC, XSum, QQP and MultiPIT, and provide original human-written documents (H-DOC) and their paraphrases (H-PP).

¹ The HLPC dataset can be found at <https://github.com/kristylht/Human-LLM-Paraphrase-Collection-HLPC>.

Table 1
Token Length Statistics of H-DOC and H-PP from the 4 corpora.

Dataset	MRPC	XSum	QQP	MultiPIT
H-DOC				
max	34	497	31	22
min	10	52	10	10
mean	23.19	269.23	13.89	12.79
H-PP				
max	34	35	35	23
min	11	4	6	6
mean	23.19	22.09	12.76	12.41

2. LLM-generated data: original LLM documents (LLM-DOC) are generated by using parts of the H-DOC documents above as prompts sent to the LLM, and their paraphrases (LLM-PP) are generated through a paraphrases that runs five paraphrasing iterations on the LLM-DOC documents.

In what follows we further elaborate to describe the data creation process in detail.

Human-written and paraphrased documents (H-DOC and H-PP). To retrieve human-generated texts alongside their paraphrases, we make use of four different datasets, namely Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), Extreme Summarization Dataset (XSum) (Narayan et al., 2018), Quora Question Pairs Dataset (QQP) (Wang et al., 2019) and Multi-Topic Paraphrases in Twitter (MultiPIT-expert) (Dou et al., 2022). MRPC consists of sentence pairs from newswire articles, while XSum dataset contains pairs of BBC articles and their summary. QQP dataset includes question pairs from Quora, and MultiPIT includes pairs of tweets from Twitter. Therefore, the combination of these data sources encompasses both clean and well-structured text data from authoritative news companies and noisy and unstructured text data from online forums. Since these datasets were published before the surge of generative AI, it is assumed that all texts are human-generated. As these datasets were originally used for paraphrase detection, pairs of sentences in these datasets are labeled as paraphrases (label = 1) or non-paraphrases of each other (label = 0), among which we are interested in those with a positive label for our purposes.

Document filtering. From the H-DOC and H-PP samples above, we remove cases of non-paraphrases, sampling only the pairs that are paraphrases to satisfy our objective of having human paraphrases for every human-generated text. Of the remaining pairs, we remove samples with fewer than 10 tokens (fewer than 30 tokens in the case of XSum due to its greater length), to enable prompt extraction for subsequent LLM text generation as described below, as well as those with more than 512 tokens due to restrictions of GPT-2 for text generation. Finally, we randomly sample 150 documents from each set, H-DOC and H-PP, with 300 documents remaining across both types. Table 1 shows the token length statistics of H-DOC and H-PP from the final filtered corpora, including the maximum, minimum and mean token length.

We next proceed to generating the LLM texts and their paraphrases. Fig. 1 illustrates the “LLM-generated documents (LLM-DOC)” and “LLM-generated paraphrases (LLM-PP)” generation process.

LLM text generation (LLM-DOC). To generate LLM-based texts that resemble the human-generated texts above, we first obtain prompts from the human-generated texts above, which are then fed to the LLM to generate new texts. The LLM-DOC generation process starts by taking the first 5 tokens from H-DOC (first 30 tokens for documents sourced from XSum) with a tokenizer, which are used as prompts for the language models in the generation process. The models then generate text based on the given prompt up to a length of the maximum length

of the H-DOC. We generate two types of LLM-DOC, watermarked and non-watermarked, both using two transformer-based language models, GPT2-XL (Solaiman et al., 2019) and OPT-1.3B (Zhang et al., 2022), along with their respective tokenizers. These models are pretrained with a wide range of internet text and therefore are suitable for text generation in this project. Default settings are used for parameter initialization for both models. Non-watermarked LLM-generated texts are output purely generated from the above two models. For watermarked LLM-DOC generation, watermarks are added by initializing a watermark in the logit processor of the models in addition to the LLM-DOC process (Kirchenbauer et al., 2023).

LLM paraphrase generation (LLM-PP). We then use the LLM-generated documents above to generate their paraphrases. For the outputs from the XSum dataset, summaries are first generated using a fine-tuned T5 model specializing in news summarization (Mishra, 2020). The output summary is limited to a length of maximum length of H-PP in the original XSum dataset. The summaries, along with the LLM-DOC generated documents from the other 3 datasets, are taken to generate LLM-PP. We use two paraphrasers, namely DIPPER and BART-paraphrase models. DIPPER is a T5-XXL paraphrase generation model with 11 billion parameters, fine-tuned on 6.3 million data points. Inherently, DIPPER is capable of capturing long-term dependencies and controlling output diversity (Krishna et al., 2024). However, due to computational resource limitations, we use a non-context version, resulting in reduced performance on long-term dependency capturing. To generate paraphrases with models that capture long-range dependencies, the second model, BART is used. The model is built upon a seq2seq architecture, with a bidirectional encoder and a unidirectional decoder (Lewis et al., 2019). The bidirectional encoder allows the model to understand sentence embeddings in a longer range, thus providing more semantic and contextual information for later paraphrase generation. Particularly, the BART-paraphrase model used in this project is a fine-tuned BART model pretrained with 3 paraphrase datasets, providing better performance in paraphrase generation. The LLM-DOCs from each dataset are passed to these paraphrasers for paraphrase (LLM-PP) generation. In order to investigate the effect of the recursive paraphrasing attack mentioned in Sadasivan et al. (2023), 5 rounds of paraphrase generation are conducted, with the 1st of paraphrases generated from the LLM-DOC, and the subsequent rounds of paraphrases generated iteratively using the outputs of each round.

Final dataset. The final HLPC dataset is composed of 600 documents, with a balanced distribution of 150 documents per type, i.e. H-DOC, H-PP, LLM-DOC and LLM-PP. These documents are grouped into two categories, human-generated (H-DOC and H-PP) and LLM-generated (LLM-DOC and LLM-PP). We use these two categories to perform binary classification in the LLM- vs human-generated text detection task, where we mix both original and paraphrased documents to evaluate their impact (particularly that of human-written paraphrases H-PP) on the detection models.

3.2. Evaluating the quality of LLM-generated paraphrases

Since paraphrases should inherently not deviate much from the original text, it is essential to evaluate the quality of the paraphrases in terms of semantic and contextual preservation. Therefore, we evaluate the semantic similarities between the original texts and their paraphrases to assess the quality of the paraphrasing, as performed in Krishna et al. (2024). Semantic similarity is chosen as the statistical method for quality assessment since it provides a more comprehensive evaluation of meaning and context on text and sentence level, compared to other possible methods, such as NER which evaluate only the name entities in the sentence. Human evaluation, as performed in Sadasivan et al. (2023) is not performed due to limited resources. We use an automated, fine-tuned sentence transformer model (Ansari, 2023) utilizing distilbert, to generate sentence embeddings for each

Table 2
Examples of LLM-generated documents.

Original Text	The dollar rose 0.6 percent to 109.54 yen (JPY =>) and climbed more than 1 percent to 1.3315 Swiss francs (CHF =).	
Non-watermarked	GPT Output	The dollar rose 0.3 percent to \$1.3240 per euro, while the euro was little changed at \$1.3240. The dollar index
	OPT Output	The dollar rose 0.2 percent against the yen to 111.85 yen, after hitting a two-week high of 111.90 yen on Friday.
Watermarked	GPT Output	The dollar rose 0.2 percent to \$1.1234 from \$1.1218, after touching a high of \$1.1218 on Friday.
	OPT Output	The dollar rose 0.1 percent against the yen to 110.65 yen, after hitting a three-month high of 110.78 yen on Tuesday.

text-paraphrase pair. We then calculate the cosine similarity between the embeddings to account for semantic similarity, scoring from -1 (least similar) to 1 (most similar). Semantic similarities are calculated between H-DOC and H-PP and between LLM-DOC and each round of LLM-PP.

Looking at human-generated data, MRPC, QQP and MultiPIT score over 0.7 for mean semantic similarity scores, indicating a good level of semantic preservation, while XSum scores only 0.383 since paraphrases in XSum are summaries of the documents. When it comes to LLM-generated data, first, the results from paraphrases generated from DIPPER and BART are compared. Fig. 2 shows the similarity scores of watermarked and non-watermarked LLM-DOC and LLM-PP. From both graphs, BART outperforms DIPPER in all datasets, particularly in MRPC and MultiPIT where BART scores over 0.9 across paraphrasing rounds which is even higher than the score from human-generated data. DIPPER's performances are generally worse than human paraphrasing, except in XSum. Meanwhile, paraphrases from both paraphrasers exhibit degradation in similarity scores across paraphrasing rounds. With recursive paraphrasing, similarity scores from BART decrease slightly from an average mean score of 0.80 in the 1st round of paraphrasing to 0.785 in the 5th round, while similarity scores of paraphrases generated by DIPPER decrease significantly from an average mean score of 0.727 in the 1st round of paraphrasing to 0.501 in the 5th round. BART's performance under recursive paraphrasing is within expectation since paraphrases generated across rounds are similar. However, the significant degradation in semantic similarity with DIPPER-generated paraphrases indicates that much semantic information is lost across paraphrasing rounds, resulting in poor-quality paraphrases that deviate from the original LLM-DOC. Second, the results from paraphrases generated from watermarked and non-watermarked LLM-DOC are compared. From Fig. 2, both semantic scores of paraphrases datasets, with average mean semantic scores of 0.732 and 0.746 , and average degradation rates of mean semantic scores of 0.168 and 0.162 respectively. Therefore, it can be concluded that watermarking does not affect the quality of paraphrases.

Lastly, we compare the results from paraphrases generated from GPT2-generated and OPT-generated documents. Paraphrases generated from GPT2-generated documents are of higher quality, with an average mean semantic score of 0.788 compared to 0.670 from OPT-generated documents. Meanwhile, the rate of quality degradation is similar for both models, with average degradation rates of mean semantic scores of 0.157 for GPT2 and 0.160 for OPT.

3.3. Descriptive statistics of HLPC

Of the four data sources that we use to build HLPC, documents and paraphrases are shorter with MRPC, QQP and MultiPIT, with

the highest mean length of 23.19 from MRPC documents, and the lowest mean length of 12.41 from MultiPIT paraphrases. XSum has longer passages as documents, having a mean length of 269.23 , and its paraphrases of 22.09 .

Looking at the LLM-generated outputs in our dataset, generally GPT2 produces outputs with longer text length, having an average mean length of 123.54 for non-watermarked outputs and 124.53 for watermarked outputs, compared to 117.95 for OPT non-watermarked outputs and 118.55 for OPT watermarked outputs. Meanwhile, watermarking does not affect text length significantly, with watermarked outputs having a slightly higher average mean length (121.54) than non-watermarked outputs (120.75). This is because watermarking does not directly decide token choices during text generation, it simply promotes the probability of certain tokens. An example is shown in Table 2 that watermarking influences the choice of tokens but poses minimal effects on text length.

We next look at the LLM paraphrases (LLM-PP). First, we observe that the text length of paraphrases decreases as the number of paraphrase rounds increases. This is more obvious from paraphrases generated by DIPPER, with a 31% decrease in average mean length from 23.82 in the 1st round of paraphrases to 16.38 in the 5th round. Paraphrases generated from BART are similar in text length across rounds of paraphrases. There is no significant difference between the text length of paraphrases generated from watermarked and non-watermarked LLM-DOC. Second, in terms of content diversity, DIPPER generates paraphrases with different choices of wordings compared to the original text while preserving the semantic information, and BART generates paraphrases that are similar or even identical to the original text. An example of recursive DIPPER- and BART-generated paraphrases of watermarked GPT output from MRPC is presented in Table 3. For more such examples, please see Appendix A.

4. LLM-generated text detection experiments

We next describe the LLM-generated text detection models we use for our experiments, as well as the evaluation metrics.

LLM-generated text detection models. For our experiments, we choose to use two SOTA models as LLM-generated text detectors, namely OpenAI RoBERTa Detector (Solaiman et al., 2019) and watermark detector (Kirchenbauer et al., 2023). The OpenAI Detector is a fine-tuned RoBERTa model trained with outputs from GPT2 model and thus is able to detect GPT2 and various LLM output text (Solaiman et al., 2019). The watermark detector classifies text by computing the number of green tokens and the probability of its existence in the given input (Kirchenbauer et al., 2023). The given input is classified as LLM-generated if the probability exceeds the set threshold.

Table 3
Examples of LLM-generated Paraphrases.

Paraphraser	ppi	Text
DIPPER	i = 0	The dollar rose 0.2 percent to \$1.1234 from \$1.1218, after touching a high of \$1.1218 on Friday.
	i = 1	The dollar rose 0.10 percent to \$1.1234 from \$1.1218, after a high of \$1.1218 on Friday
	i = 2	The dollar rose 0.10 percent to \$1.1234 from \$1.1218.
	i = 3	The dollar rose by a penny to \$1.1234 from \$1.1218
	i = 4	The dollar jumped a penny to \$1.1234 from \$1.1218.
	i = 5	A little more, the dollar was up a penny to \$1.1234.
BART	i = 0	The dollar rose 0.2 percent to \$1.1234 from \$1.1218, after touching a high of \$1.1218 on Friday.
	i = 1	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
	i = 2	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
	i = 3	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
	i = 4	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
	i = 5	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.

Using the HLPC dataset, we test models on a balanced set of 600 documents, 300 human-generated and 300 LLM-generated. For testing involving LLM-PP, the experiment is repeated 5 times, using each of the 5 rounds of AP. The parameters of the watermark detector are set according to the parameters used in watermark AI document generation for effective classification. OpenAI Detector is used on non-watermarked LLM-DOC and their paraphrases, while the watermark detector on watermarked LLM-DOC and their paraphrases.

Experiment settings. The experiment is repeated with the outputs from different combinations of the 4 datasets (MRPC, XSum, QQP and MultiPIT), 2 generative language models (GPT and OPT) and 2 paraphrasers (DIPPER and BART). The final classification is conducted with the full set of human-generated data and LLM-generated data, which means that 150 samples are taken from each of the documents and paraphrases, resulting in a total sample size of 600 for classification.

Evaluation metrics. To evaluate the performance of the LLM text detectors, we compute the accuracy, TPR@1%FPR and AUROC for each round of classification. These metrics are chosen as they are widely used in the literature, and they provide a comprehensive analysis of the classification performance. Specifically, AUROC provides an overview of the tradeoff between the true positive rate (correctly classifying LLM-generated data) and false positive rate (misclassifying human-generated data as LLM-generated), and TPR@1%FPR shows the performance of the classifiers focused on improving the True Positive Rate (TPR) while also trying to keep the False Positive Rate (FPR) low.

The equations of AUROC, TPR and FPR are as follows:

$$\begin{aligned} \text{TPR} &= \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \\ \text{FPR} &= \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}} \end{aligned} \quad (1)$$

$$\text{AUROC} = \int_0^1 \text{TPR}(t) d\text{FPR}(t) \quad (2)$$

TPR@1%FPR is the TPR while FPR is set to a 1% threshold. To investigate the effects of H-PP in LLM-generated text classification, comparisons of classifiers' performance are made based on the difference in the above statistics from various text pairs. Results between LLM-DOC and LLM-PP when paired with H-DOC and H-PP are compared.

5. Results & analysis

5.1. Classification with human-generated data and LLM-generated documents

We perform two classification experiments to observe the effects of including H-PP on LLM-generated text detection. The first classification is conducted with H-DOC and LLM-DOC, while the second classification with H-PP and LLM-DOC.

Fig. 3(a) shows the ROC curve of the results with non-watermarked and watermarked LLM-DOC respectively. First, the results of classifications in terms of non-watermarked and watermarked LLM-DOC are evaluated. For both non-watermarked and watermarked LLM-DOC, the results are satisfactory either with or without H-PP, with over 75% of classifications scoring an AUROC > 0.85. Among them, the results from watermarked LLM-DOC are better than the results from non-watermarked LLM-DOC, with 87.5% of classification scoring an AUROC > 0.85, compared to 62.5% from non-watermarked LLM-DOC. This shows that watermarking is a more effective strategy in LLM-generated text detection.

Figs. 3(a) and 3(b) show that the inclusion of H-PP generally decreases AUROC and accuracy and increases TPR@1%FPR, compared to the results with H-DOC. Among all the data sources, results from Xsum show the highest decrease of 0.142 AUROC and 0.153 accuracy and the highest increase of 0.24 TPR@1%FPR. This implies that H-PP might contain similar semantic and contextual information to LLM-DOC, making it more challenging for the model to distinguish between the classes. However, the increase in TPR@1%FPR indicates the identification of LLM-DOC is promoted while ensuring a low percentage of misclassification of human-generated data as LLM-generated (False Positive) with H-PP.

In addition, **Figs. 3(c) and 3(d)** show a different phenomenon while including H-PP with watermarked LLM-DOC, with the inclusion of H-PP generally increasing AUROC, TPR@1%FPR, and posing no effects on accuracy. Among all the data sources, Xsum shows the highest increase of 0.046 AUROC, and QQP shows the highest increase of 0.234 TPR@1%FPR. This implies that model performances are promoted while H-PP is included with watermarked LLM-DOC. Overall, the results show the effects of including H-PP are highly dependent on the types of LLM-DOC used in classification. While it increases TPR@1%FPR in all scenarios, it also decreases AUROC and accuracy when non-watermarked LLM-DOC are used.

5.2. Classification under recursive paraphrasing

We next evaluate the effects of including H-PP with recursive paraphrases. In the interest of brevity and focus, we analyze results for MRPC GPT-generated text here, but provide full details for all datasets in [Appendix B](#), whose results are consistent with MRPC. The results (H-PP_pp0, H-PP_pp2 & H-PP_pp5) are compared to the classification result without the use of H-PP (pp0, pp2 & pp5). First, the general model performance is discussed in terms of the differences between DIPPER and BART-generated paraphrases.

Fig. 4 shows that regardless of the presence of watermarking and H-PP, model performances with DIPPER-generated paraphrases degrade to a larger extent than with BART-generated paraphrases across rounds of paraphrasing. A decrease of 0.26 in average AUROC and 0.67 in average TPR@1%FPR are observed from DIPPER, while BART's average AUROC and TPR@1%FPR decrease merely 0.0075 and 0.045 respectively. Particularly, the classification with DIPPER-generated paraphrases from watermarked LLM-DOC (**Fig. 4(c)**) shows the highest

Table 4

Average results of classification with full human-written and LLM-generated data; bracketed numbers indicate percentage difference.

LLM-generated Documents	Paraphraser	Data	AUROC	TPR@1%FPR	Accuracy
Non-watermarked	DIPPER	pp1	0.908	0.285	0.814
		hp_pp1	0.883	0.358	0.769
		f_pp1	0.884 (-2.71%)	0.379 (32.98%, 5.86%)	0.778 (-4.62%)
		pp5	0.878	0.138	0.811
		hp_pp5	0.843	0.169	0.766
		f_pp5	0.867 (-1.27%)	0.290 (110.14%, 71.60%)	0.775 (-2.32%)
	BART	pp1	0.873	0.209	0.793
		hp_pp1	0.839	0.284	0.750
		f_pp1	0.874 (0.001%, 4.17%)	0.313 (49.76%, 10.21%)	0.771 (-2.85%)
		pp5	0.876	0.201	0.795
		hp_pp5	0.842	0.276	0.751
		f_pp5	0.874 (-0.002%)	0.341 (69.65%, 23.55%)	0.773 (-2.85%)
Watermarked	DIPPER	pp1	0.652	0.100	0.508
		hp_pp1	0.678	0.117	0.508
		f_pp1	0.636 (-6.60%)	0.209 (109%, 78.63%)	0.556 (9.45%, 9.45%)
		pp5	0.512	0.019	0.500
		hp_pp5	0.541	0.022	0.500
		f_pp5	0.587 (14.65%, 8.5%)	0.185 (873.68%, 740.91%)	0.554 (10.8%, 10.8%)
	BART	pp1	0.825	0.445	0.611
		hp_pp1	0.840	0.467	0.611
		f_pp1	0.706 (-18.98%)	0.315 (-48.25%)	0.584 (-4.62%)
		pp5	0.797	0.398	0.599
		hp_pp5	0.813	0.426	0.598
		f_pp5	0.700 (-16.14%)	0.307 (-38.76%)	0.581 (-3.10%)

degradation in model performance, resulting in the lowest AUROC of 0.49 which is worse than a random classifier. This can be attributed to the lower semantic similarity between the LLM-DOC and DIPPER-generated LLM-PP mentioned in Section 3.2. The low semantic similarity might indicate that DIPPER-generated LLM-PP becomes more similar to H-PP coincidentally while deviating from the original LLM-DOC, resulting in a degradation in model performance.

In addition, classification with BART-generated LLM-PP (Fig. 4(d)) shows excellent results with AUROC 0.98 even after 5 rounds of paraphrasing. Second, the effects of including H-PP in classification are evaluated. In both classifications with paraphrases from watermarked and non-watermarked LLM-DOC, the performances of the models with H-PP are better than those without H-PP, while the effect of the promotion is more significant with paraphrases generated from non-watermarked LLM-DOC. For classification with paraphrases generated from non-watermarked LLM-DOC and H-PP (Figs. 4(a) and 4(b)), TPR@1%FPR increases from a minimum of 0.02 to a maximum of 0.153 when compared to classification with H-DOC. The positive effect is less significant on AUROC with an average increase of 0.0028. Although minimal effects are cast on AUROC and accuracy with the use of H-PP, the significant increase in TPR@1%FPR shows that H-PP helps in the detection of LLM-generated data at a low false positive rate, ensuring that less human data are classed as LLM-generated.

For classification with paraphrases generated from watermarked LLM-DOC and H-PP (Figs. 4(c) and 4(d)), minimal increases in AUROC and TPR@1%FPR are observed, with the highest increase in AUROC of 0.019 and TPR@1%FPR of 0.02. Overall, it can be concluded that including H-PP in the classification under recursive paraphrasing helps promoting AUROC and TPR@1%FPR under recursive paraphrasing. The comparison of classification results between the inclusion and exclusion of H-PP with other datasets and generative models are also presented in Appendix B. Generally, similar results are observed, except for classification with Xsum where the inclusion of H-PP reduces AUROC.

5.3. Classification with full set of human-generated data and LLM-generated data

We now perform classification experiments with the full set of human-written and LLM-generated data, passing both documents and paraphrases to the classifier.

Table 4 shows the comparison of the average results from classification with (i) H-DOC vs LLM-PP (pp1 & pp5), (ii) H-PP vs LLM-PP (H-PP_pp1 & H-PP_pp5) and (iii) full human data vs LLM-generated data (f_pp1 & f_pp5) across all datasets. First, for results from paraphrases generated from watermarked LLM-DOC, model performance shows extreme opposites depending on the paraphrasers used. The best improvement in statistical results is shown using DIPPER-generated paraphrases from watermarked LLM-DOC, along with H-DOC, H-PP and LLM-DOC (f_pp1 & f_pp5). With the 1st round of paraphrases, TPR@1%FPR increases by 109% and 78.63% and accuracy increases by 9.45% and 9.45%, compared to only using H-DOC or H-PP respectively. With the 5th round of paraphrases, AUROC increases by 14.65% and 8.5%, TPR@1%FPR increases by 874% and 740% and accuracy increases by 10.8% and 10.8%, compared to only using H-DOC or H-PP respectively. This shows that using the full set of data as input for classification is significantly effective in improving LLM-generated text detection under recursive paraphrasing, with the condition that the LLM-DOC is watermarked and paraphrases are DIPPER-generated. Meanwhile, the worst performance is shown with BART-generated paraphrases from watermarked LLM-DOC under the same condition of using the full set of data as input. Compared with the best results, results from 1st and 5th rounds of BART-generated paraphrases show a degradation of 18.98% and 16.14% in AUROC, 48.25% and 38.76% in TPR@1%FPR and 4.62% and 3.10% in accuracy respectively. The significant difference in the statistical results shows that the watermark detector is highly sensitive to the paraphraser used. While the performance improves significantly with DIPPER-generated paraphrases, it also degrades significantly with BART-generated paraphrases.

Second, TPR@1%FPR significantly increases while using paraphrases generated from non-watermarked LLM-DOC along with H-DOC, H-PP and LLM-DOC, while AUROC and accuracy remain unchanged or decrease slightly. Results show that TPR@1%FPR increases by a minimum of 5.86% to a maximum of 110.14%, with an average of 42.84% across different inputs. This indicates that having the full set of data as input effectively improves TPR@1%FPR and ensures minimal misclassification at a low false positive rate. Meanwhile, AUROC and accuracy decrease, compared to the results from H-DOC vs LLM-PP and H-PP vs LLM-PP. However, the decrease is less significant compared to the increase in TPR@1%FPR. The maximum decrease in AUROC and accuracy is merely 4.17% and 4.62% respectively. Therefore, it can be concluded that having a full set of data as input exhibits a trade-off between accuracy, AUROC and TPR@1%FPR. Considering

the significant improvement in TPR@1%FPR and the importance of ensuring minimal misclassification, having a full set of data as input is suggested to be a better method than using merely H-DOC under recursive paraphrasing. Detailed ROC curve and statistical results of classification with each dataset and its generations are presented in [Appendix C](#).

5.4. Benefits of using human-written paraphrases in LLM or detectors training

In our review of the literature in Section 2 we found that existing LLMs are trained on corpora that do not contain H-PP information. Since existing detectors are designed to identify the LLM-generated statistical pattern and watermark from the input text, paraphrasing which reduces or erases the above characteristics could effectively evade the detectors. Our experiments show that including H-PP in the dataset promotes classification performances under different circumstances, and therefore, including H-PP in the training datasets during the training of detectors could effectively improve detectors' classification performances since models could learn about the fundamental differences of semantic and contextual information between human-written and LLM-generated text, even under recursive paraphrasing.

Our results also show that the effectiveness of including H-PP in the dataset is highly dependent on the existence of watermarking and the type of paraphraser used. In our experiments, while H-DOC is included for classification, watermarking and DIPPER-generated paraphrases help improve classification performance, while experiments with non-watermarked and BART-generated paraphrases show the opposite. As such, it is important to understand and predict the potential usage of watermarking and the type of paraphraser while developing the detectors. Detector developers could either get the information from users or employ a multi-step classification model for accurate prediction. A multi-step classification model could first identify the presence of watermark and the type of paraphrasers, then decide whether to include H-PP in the training dataset of the detectors based on the results. If such technology or information is not available, it is recommended to include either H-DOC or H-PP, to avoid significant degradation in classification performance. Meanwhile, we show that including H-PP in the datasets is highly effective under recursive paraphrasing. As such, we recommend that detectors, which are used in circumstances where paraphrasing is prevalent, for example, in academic publications, to be trained with H-PP instead of only H-DOC, so as to increase AUROC and TPR@1%FPR.

6. Conclusion

In this study, our aim was to investigate the effect of human paraphrases (H-PP) on LLM-generated text detection by conducting classifications with various combinations of human and LLM-generated data pairs. To enable this study, we devise a data collection strategy and generate the HLPC dataset by leveraging and extending four existing data sources: MRPC, XSum, QQP and MultiPIT. Unlike previous datasets, our new dataset, Human & LLM Paraphrase Collection (HLPC), incorporates human-written documents (H-DOC), human-written paraphrases (H-PP), LLM-generated texts (LLM-DOC) and LLM-generated paraphrases (LLM-PP). We generate LLM documents by prompting GPT2-XL and OPT-13B with prompts derived from human-written documents. AI paraphrasers DIPPER and BART are then used to paraphrase the generated outputs. Using this dataset, we perform classification experiments with SOTA LLM-generated text detectors OpenAI RoBERTa and watermark detector, with the aim of understanding the effects of incorporating human-written paraphrases in LLM-generated text detection. Data pairs used for classifications include (i) H-DOC vs LLM-DOC, (ii) H-PP vs LLM-DOC, (iii) H-DOC vs LLM-PP, (iv) H-PP vs LLM-PP and (v) H-DOC & H-PP vs LLM-DOC & LLM-DOC. 3 comparisons are made between the classification results to examine the effects of including

H-PP in classification. First, results from (i) and (ii) are compared to show H-PP's effects while classification is done with LLM-DOC. Second, results from (iii) and (iv) are compared to show H-PP's effects under recursive paraphrasing. Lastly, results from (v) are compared to results from (iii) and (iv) to examine the effects of having a full set of human and LLM-generated data.

In our experiments, we observe that in all 3 sets of comparisons, including H-PP in the classification is effective in promoting TPR@1%FPR, while its effects on AUROC and accuracy are highly dependent on the presence of watermarking and the type of paraphraser. In the 1st set of comparison, the results show that TPR@1%FPR increases in all scenarios, but AUROC and accuracy decrease if non-watermarked LLM-DOC are used. For the 2nd set of comparison, AUROC and TPR@1%FPR increases to a small extent in all scenarios, while accuracy remains unchanged under recursive paraphrasing. Lastly, for the 3rd set of comparison with the full set of data, results vary in 2 extremes depending on the paraphraser used to generate paraphrases from watermarked LLM-DOC, while TPR@1%FPR increase significantly and AUROC and accuracy decrease slightly with non-watermarked LLM-DOC and their paraphrases. Therefore, it can be concluded that the inclusion of H-PP in classification promotes TPR@1%FPR with a possible trade-off of AUROC and accuracy.

Our study has potential to be further extended in the future by studying additional datasets and detection models, to tackle some of the limitations of our study. First, the sentences in the chosen datasets are relatively short, with a mean token length of 79.78 for H-DOC and 17.61 for H-PP. Since the performance of LLM-generated text detectors increases with the input text length, consideration of additional datasets with longer sentences would help provide a more diverse analysis of the effects of H-PP's inclusion in classification. However, due to the limited availability of datasets that contain H-PP, only datasets with short sentences are used in this project. Second, regarding the quality assessment of LLM-generated paraphrases, additional metrics and embedding models can be utilized, such as "Ada Score" and "SimCSE Score" calculated in [Jayawardena et al. \(2024\)](#), in order to evaluate the quality of LLM-generated paraphrases in terms of different embedding models. Third, when it comes to the generation of paraphrases using LLMs, we used the default settings. Future work can explore other variants by generating paraphrases with different decoding strategies and parameters of the LLM paraphrasers, with the aim of further examining their effects on semantic similarity and detector performance. Lastly, other SOTA LLM text detection tools could be tested to broaden the findings, such as GPTZero,² which was excluded from our study due to its associated costs.

CRediT authorship contribution statement

Hiu Ting Lau: Writing – original draft, Resources, Methodology, Formal analysis, Data curation, Conceptualization. **Arkaitz Zubiaga:** Writing – review & editing, Validation, Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Examples of LLM-generated paraphrases

See [Table 5](#).

² <https://gptzero.me/>

Table 5Examples of LLM-generated paraphrases; ppi means the i th round of paraphrase.

MRPC			
Paraphraser	Input	ppi	Text
DIPPER	Watermarked GPT Output	i = 0	The dollar rose 0.2 percent to \$1.1234 from \$1.1218, after touching a high of \$1.1218 on Friday.
		i = 1	The dollar rose 0.10 percent to \$1.1234 from \$1.1218, after a high of \$1.1218 on Friday
		i = 2	The dollar rose 0.10 percent to \$1.1234 from \$1.1218.
		i = 3	The dollar rose by a penny to \$1.1234 from \$1.1218.
		i = 4	The dollar jumped a penny to \$1.1234 from \$1.1218.
		i = 5	A little more, the dollar was up a penny to \$1.1234.
DIPPER	Watermarked OPT Output	i = 0	The dollar rose 0.1 percent against the yen to 110.65 yen, after hitting a three-month high of 110.78 yen on Tuesday.
		i = 1	The dollar slipped a quarter of a percent against the yen, after hitting a three-month high of 110.65 yen on Tuesday
		i = 2	The yen slipped a quarter of a percent against the dollar, after hitting a three-month high of 110.60.
		i = 3	It is a good thing that it is not going to be long. The yen is a little less than one yen, the
		i = 4	But the yen is a little less than one yen. The yen is just a tenth of a yen.
		i = 5	This is the amount of ten thousand yen, but the yen is only ten thousand yen.
DIPPER	Non-watermarked GPT Output	i = 0	The dollar rose 0.3 percent to \$1.3240 per euro, while the euro was little changed at \$1.3240. The dollar index
		i = 1	The dollar was gaining 0.3 percent to \$1.3240, and the Euro was less than \$1,400.
		i = 2	The dollar was up 0.3 percent to \$1.3240. The Euro was less than \$1,400.
		i = 3	It was a little over \$2,000. The Euro was not much more than \$3,000.
		i = 4	The Euro was not much more than a thousand rubles.
		i = 5	The Euro was less than a thousand rubles.
DIPPER	Non-watermarked OPT Output	i = 0	The dollar rose 0.2 percent against the yen to 111.85 yen, after hitting a two-week high of 111.90 yen on Friday.
		i = 1	On Friday the yen rose by 0.2 percent against the yen. The yen fell by 0.2 percent against the dollar. The pair had hit
		i = 2	The pair had fallen to the tenth day of the previous month. On Friday the yen rose by 0.2 percent against the dollar.
		i = 3	On Friday the yen rose by 0.2 percent against the dollar. The pair had fallen to the tenth day of the previous month.
		i = 4	But if you were to use the yen as an example, you would have a hard time keeping up the momentum. On Friday, the dollar fell by
		i = 5	But you would have to think of the yen as an example, and I doubt if you would be able to sustain the momentum.
BART	Watermarked GPT Output	i = 0	The dollar rose 0.2 percent to \$1.1234 from \$1.1218, after touching a high of \$1.1218 on Friday.
		i = 1	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
		i = 2	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
		i = 3	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
		i = 4	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
		i = 5	The dollar rose 0.2 percent to \$1.1234 on the New York Stock Exchange, after touching a high of \$1.1218 on Friday.
BART	Watermarked OPT Output	i = 0	The dollar rose 0.1 percent against the yen to 110.65 yen, after hitting a three-month high of 110.78 yen on Tuesday.
		i = 1	The dollar rose 0.1 percent against the yen to 110.65 yen, after hitting a three-month high of 110.78 yen on Tuesday.
		i = 2	The dollar rose 0.1 percent against the yen to 110.65 yen, after hitting a three-month high of 110.78 yen on Tuesday.
		i = 3	The dollar rose 0.1 percent against the yen to 110.65 yen, after hitting a three-month high of 110.78 yen on Tuesday.
		i = 4	The dollar rose 0.1 percent against the yen to 110.65 yen, after hitting a three-month high of 110.78 yen on Tuesday.
		i = 5	The dollar rose 0.1 percent against the yen to 110.65 yen, after hitting a three-month high of 110.78 yen on Tuesday.
BART	Non-watermarked GPT Output	i = 0	The dollar rose 0.3 percent to \$1.3240 per euro, while the euro was little changed at \$1.3240. The dollar index
		i = 1	The dollar rose 0.3 percent against the euro to \$1.3240, while the euro was little changed against the dollar at 1.3243.
		i = 2	The dollar rose 0.3 percent against the euro to \$1.3240, while the euro was little changed against the dollar at 1.3243.
		i = 3	The dollar rose 0.3 percent against the euro to \$1.3240, while the euro was little changed against the dollar at 1.3243.
		i = 4	The dollar rose 0.3 percent against the euro to \$1.3240, while the euro was little changed against the dollar at 1.3243.
		i = 5	The dollar rose 0.3 percent against the euro to \$1.3240, while the euro was little changed against the dollar at 1.3243.
BART	Non-watermarked OPT Output	i = 0	The dollar rose 0.2 percent against the yen to 111.85 yen, after hitting a two-week high of 111.90 yen on Friday.
		i = 1	The dollar rose 0.2 percent against the yen to 110.85 yen, after hitting a two-week high of 110.90 yen on Friday.
		i = 2	The dollar rose 0.2 percent against the yen to 110.85 yen, after hitting a two-week high of 110.90 yen on Friday.
		i = 3	The dollar rose 0.2 percent against the yen to 110.85 yen, after hitting a two-week high of 110.90 yen on Friday.
		i = 4	The dollar rose 0.2 percent against the yen to 110.85 yen, after hitting a two-week high of 110.90 yen on Friday.
		i = 5	The dollar rose 0.2 percent against the yen to 110.85 yen, after hitting a two-week high of 110.90 yen on Friday.

Appendix B. Classification results with human-written document/paraphrases and LLM-generated paraphrases**Appendix C. Classification results with full human-written data and LLM-generated data**

See Figs. 5–18.

See Figs. 19–34.

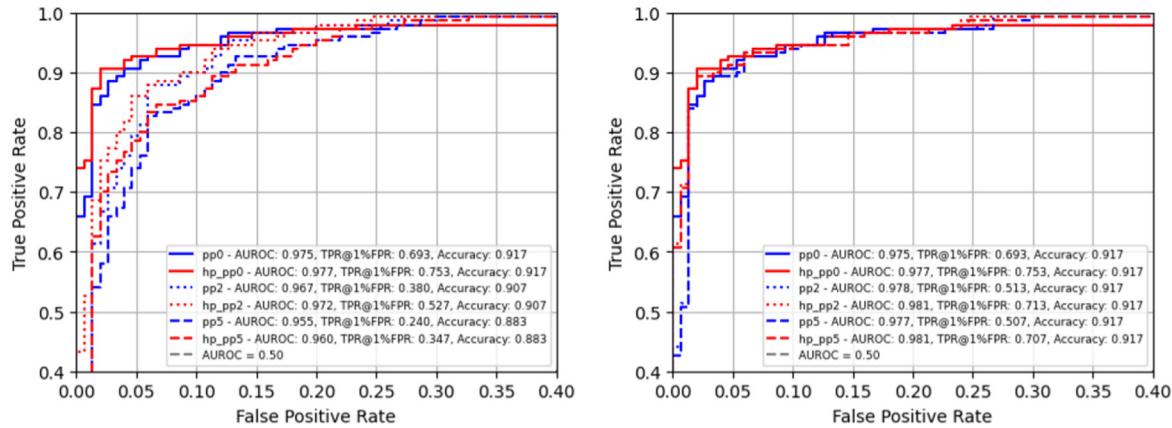


Fig. 5. ROC Curve from OpenAI Detector with Human-generated Documents/Paraphrases and Non-watermarked DIPPER (left) / BART-generated (right) Paraphrases from MRPC OPT-Generated Text.

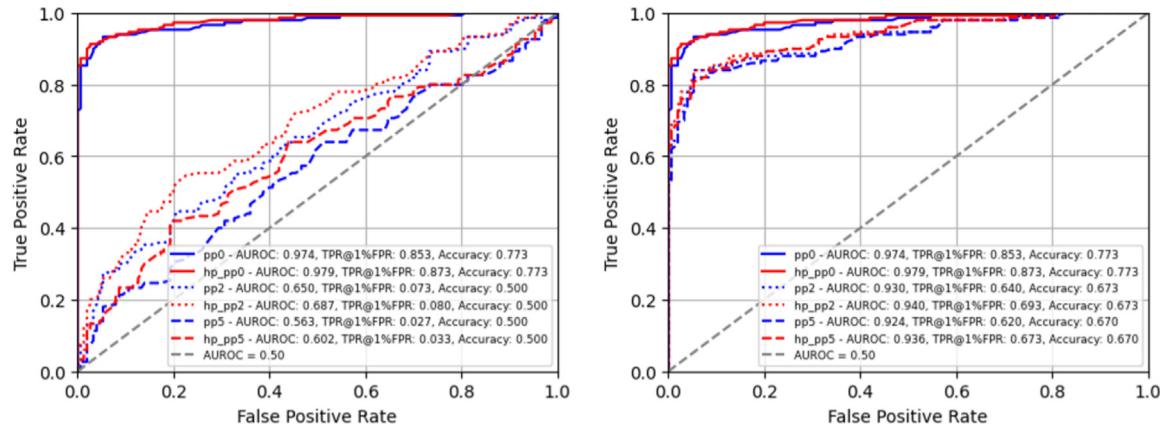


Fig. 6. ROC Curve from Watermark Detector with Human-generated Documents/Paraphrases and Watermarked DIPPER (left) / BART-generated (right) Paraphrases from MRPC OPT-Generated Text.

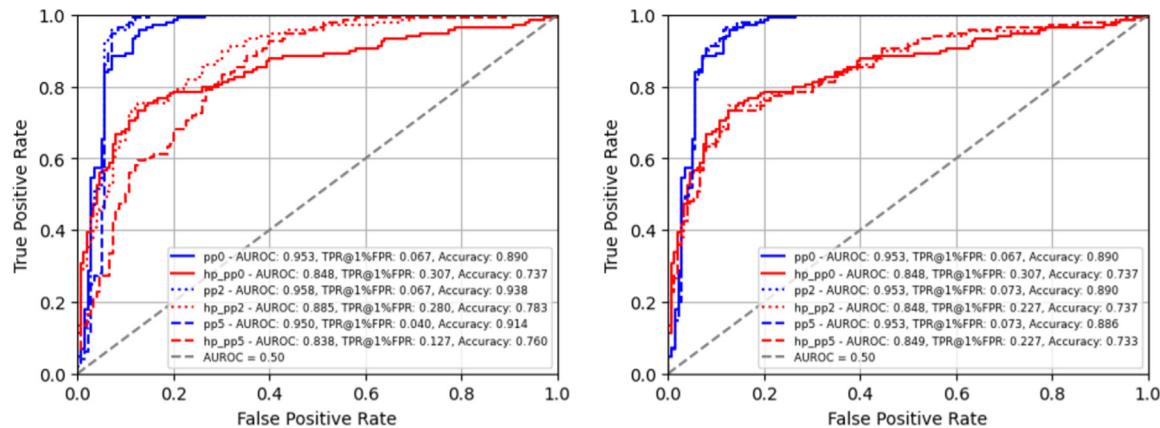


Fig. 7. ROC Curve from OpenAI Detector with Human-generated Documents/Paraphrases and Non-watermarked DIPPER (left) / BART-generated (right) Paraphrases from XSum GPT-Generated Text.

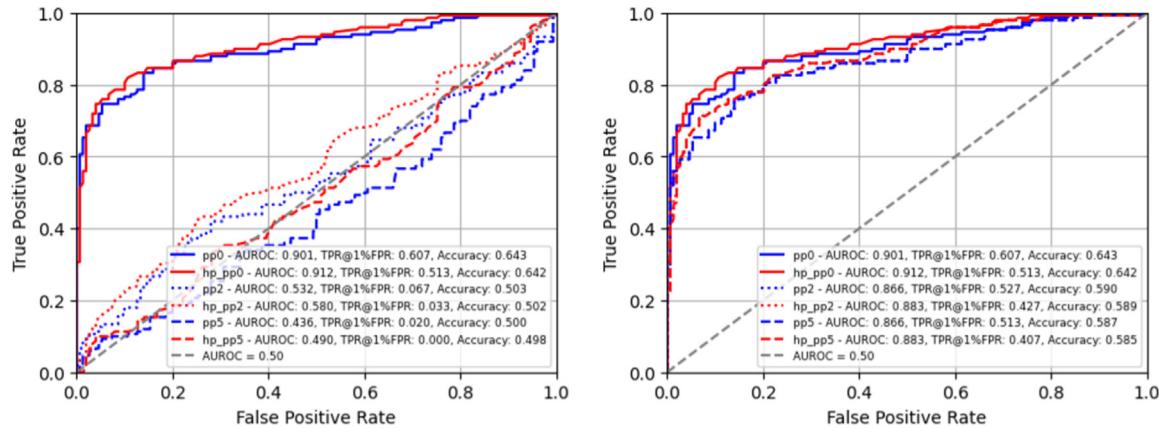


Fig. 8. ROC Curve from Watermark Detector with Human-generated Documents/Paraphrases and Watermarked DIPPER (left) / BART-generated (right) Paraphrases from XSum GPT-Generated Text.

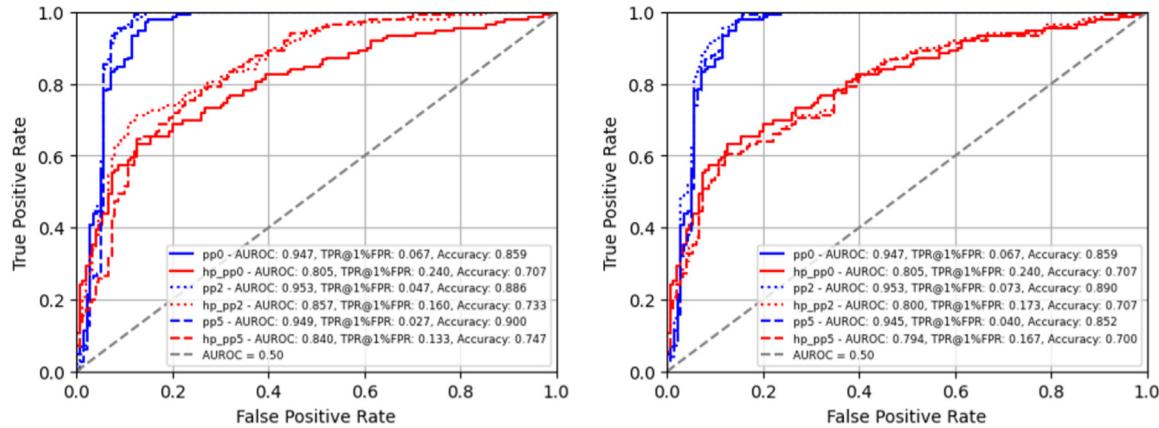


Fig. 9. ROC Curve from OpenAI Detector with Human-generated Documents/Paraphrases and Non-watermarked DIPPER (left) / BART-generated (right) Paraphrases from XSum OPT-Generated Text.

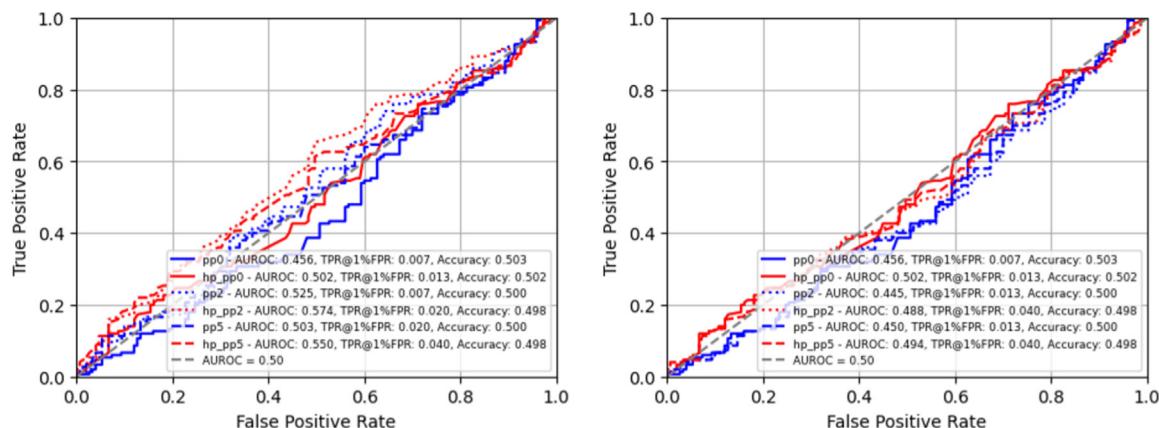


Fig. 10. ROC Curve from Watermark Detector with Human-generated Documents/Paraphrases and Watermarked DIPPER (left) / BART-generated (right) Paraphrases from XSum OPT-Generated Text.

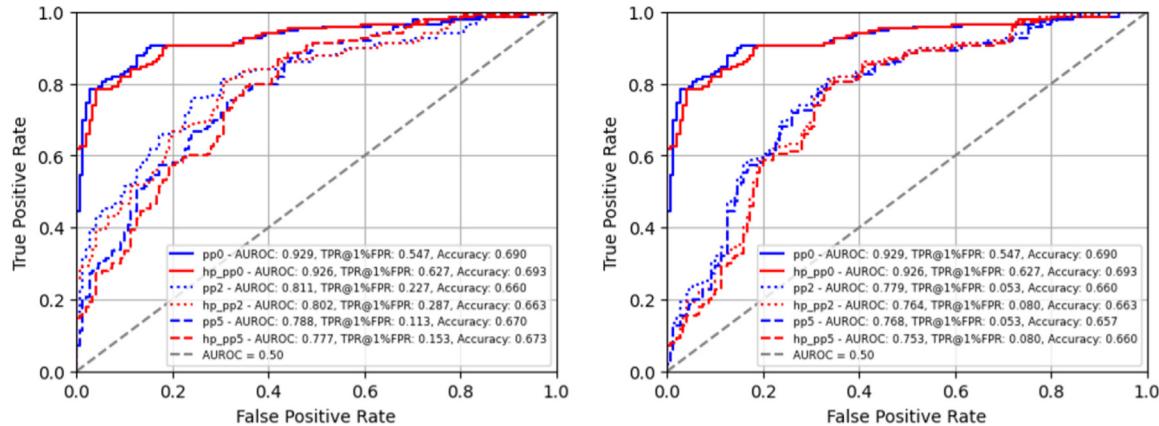


Fig. 11. ROC Curve from OpenAI Detector with Human-generated Documents/Paraphrases and Non-watermarked DIPPER (left) / BART-generated (right) Paraphrases from QQP GPT-Generated Text.

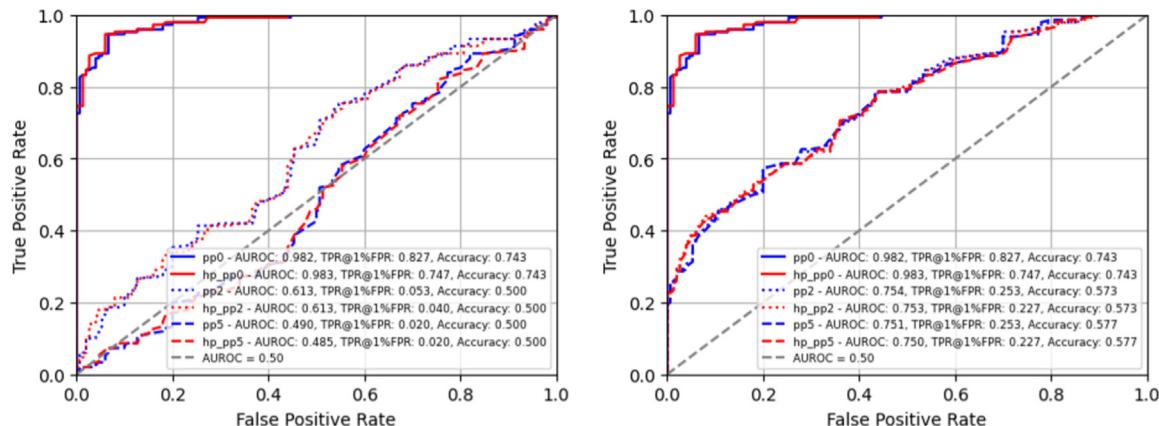


Fig. 12. ROC Curve from Watermark Detector with Human-generated Documents/Paraphrases and Watermarked DIPPER (left) / BART-generated (right) Paraphrases from QQP GPT-Generated Text.

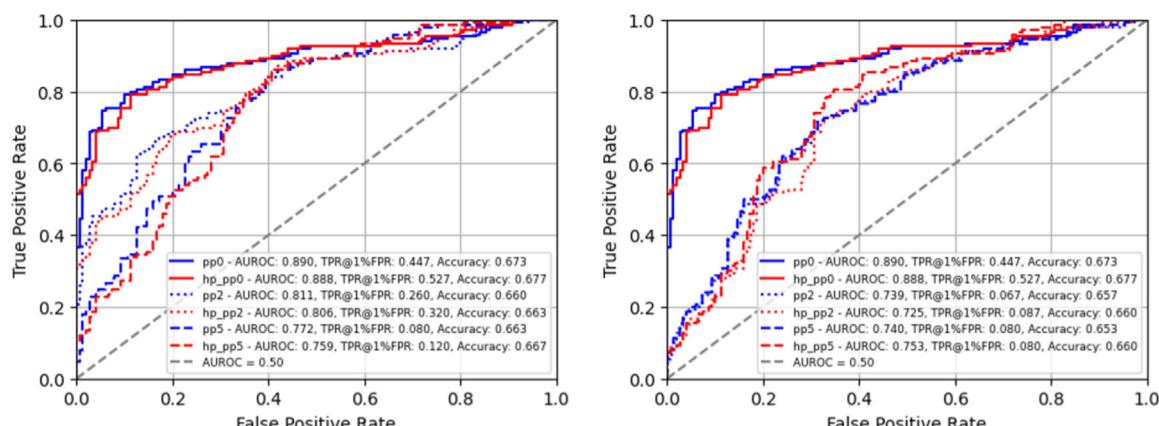


Fig. 13. ROC Curve from OpenAI Detector with Human-generated Documents/Paraphrases and Non-watermarked DIPPER (left) / BART-generated (right) Paraphrases from QQP OPT-Generated Text.

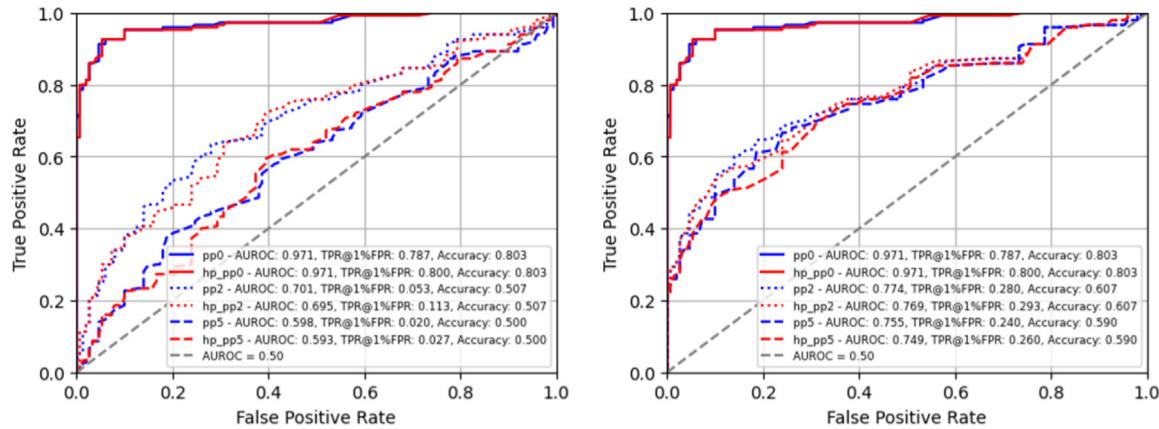


Fig. 14. ROC Curve from Watermark Detector with Human-generated Documents/Paraphrases and Watermarked DIPPER (left) / BART-generated (right) Paraphrases from QQP OPT-Generated Text.

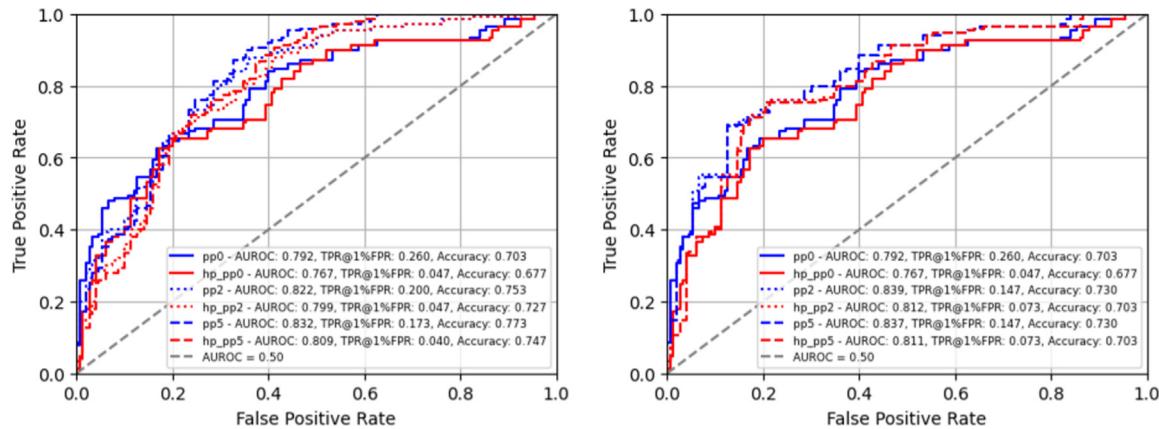


Fig. 15. ROC Curve from OpenAI Detector with Human-generated Documents/Paraphrases and Non-watermarked DIPPER (left) / BART-generated (right) Paraphrases from MultiPIT GPT-Generated Text.

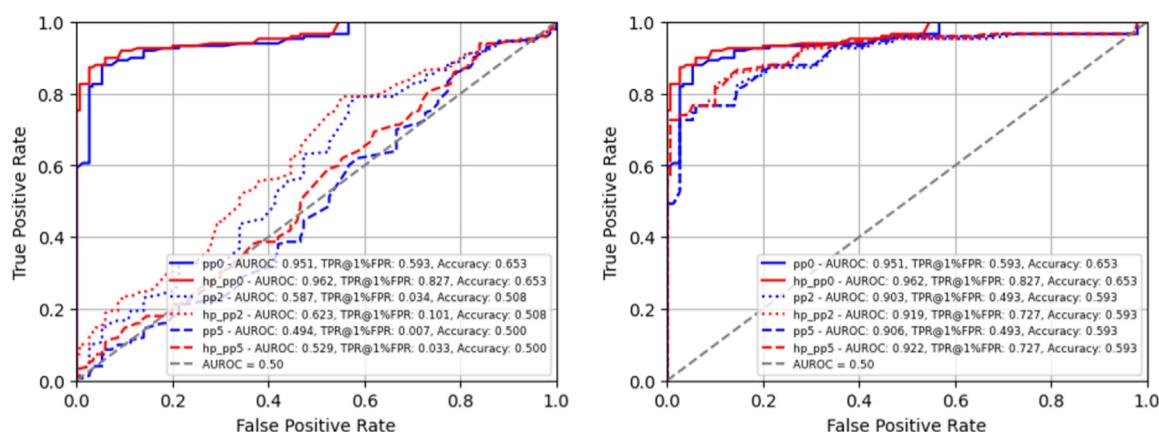


Fig. 16. ROC Curve from Watermark Detector with Human-generated Documents/Paraphrases and Watermarked DIPPER (left) / BART-generated (right) Paraphrases from MultiPIT GPT-Generated Text.

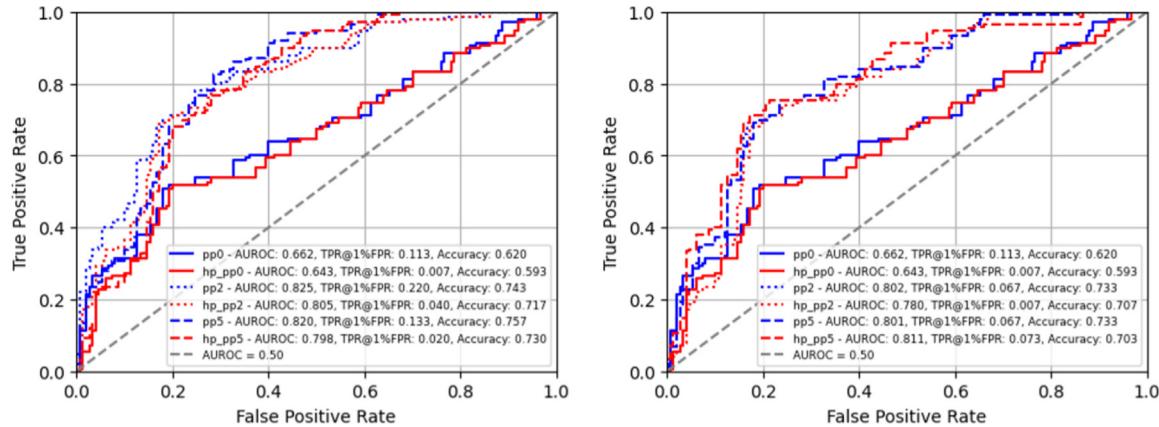


Fig. 17. ROC Curve from OpenAI Detector with Human-generated Documents/Paraphrases and Non-watermarked DIPPER (left) / BART-generated (right) Paraphrases from MultiPIT OPT-Generated Text.

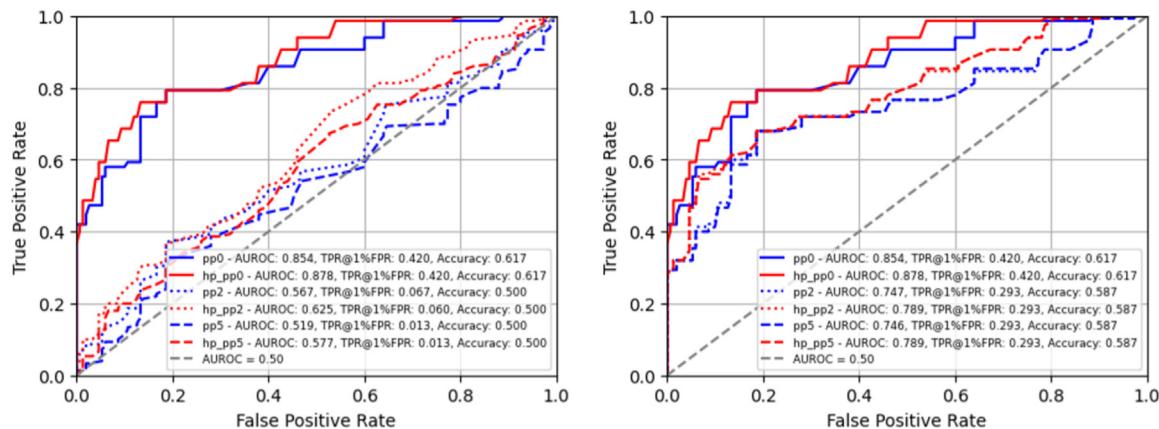


Fig. 18. ROC Curve from Watermark Detector with Human-generated Documents/Paraphrases and Watermarked DIPPER (left) / BART-generated (right) Paraphrases from MultiPIT OPT-Generated Text.

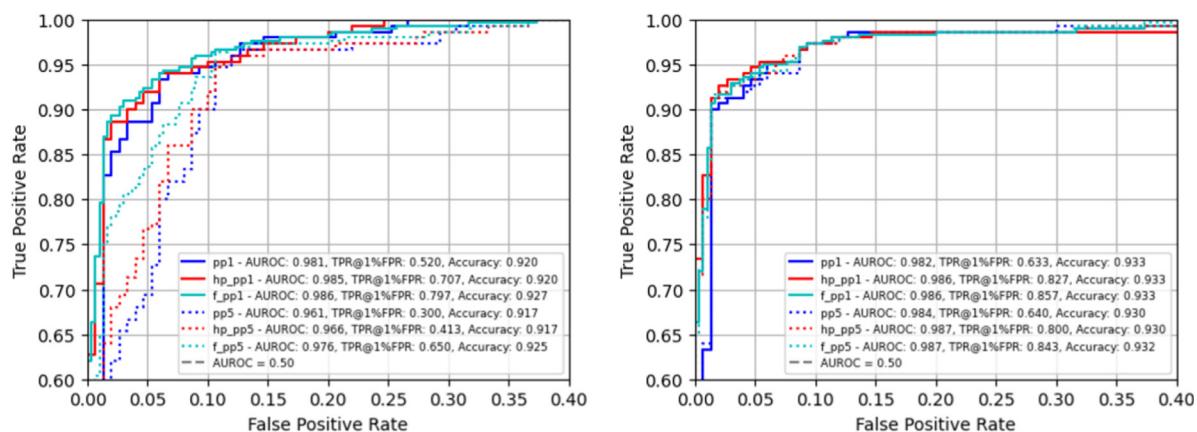


Fig. 19. ROC Curve with Full Human Data and Non-watermarked MRPC LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from MRPC GPT2-generated Text).

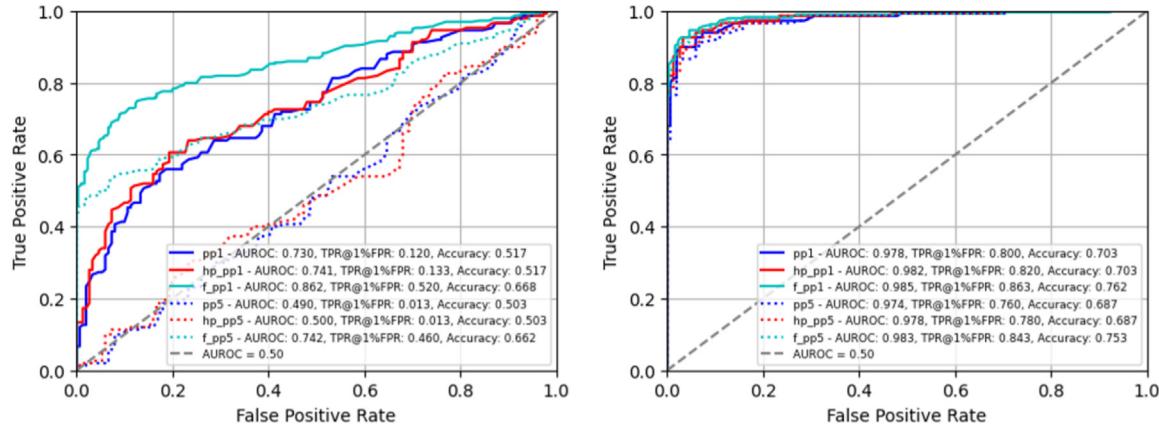


Fig. 20. ROC Curve with Full Human Data and Watermarked MRPC LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from MRPC GPT2-generated Text).

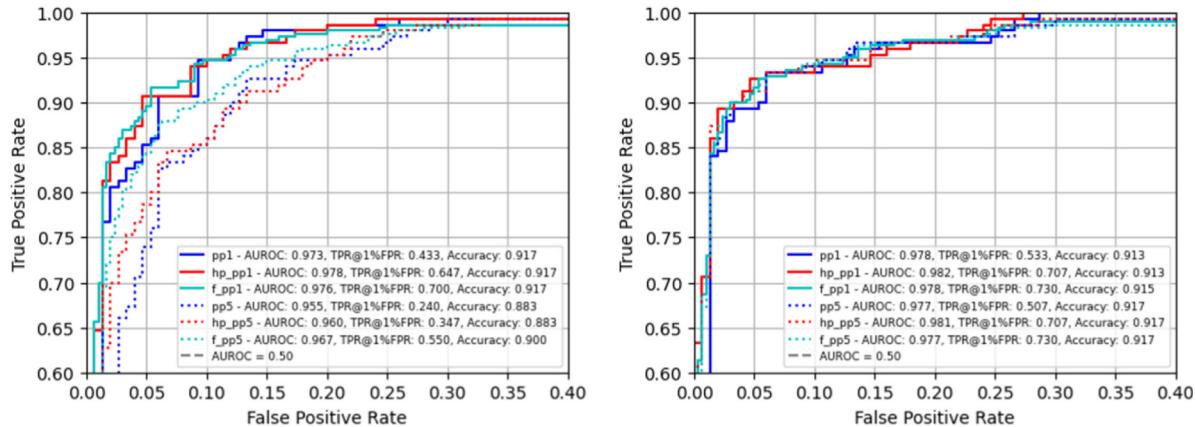


Fig. 21. ROC Curve with Full Human Data and Non-watermarked MRPC LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from MRPC OPT-generated Text).

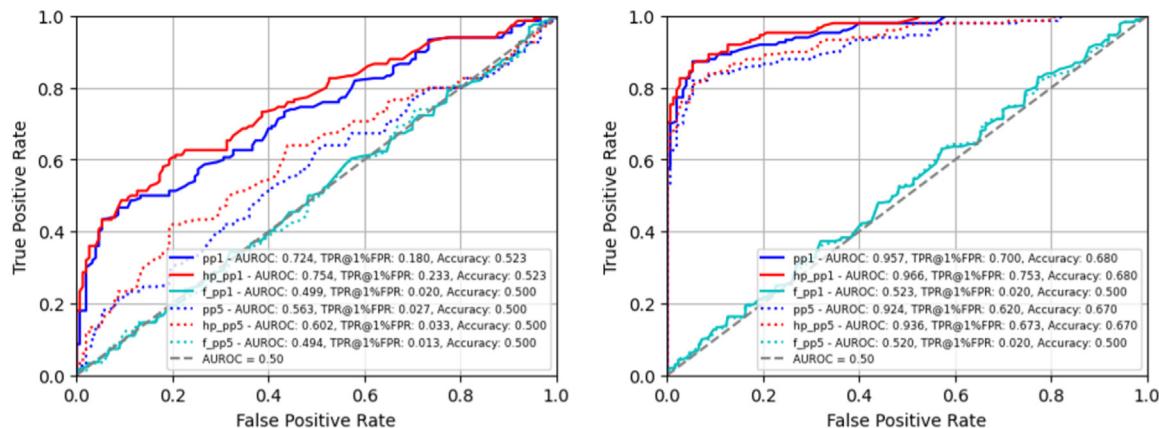


Fig. 22. ROC Curve with Full Human Data and Watermarked MRPC LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from MRPC OPT-generated Text).

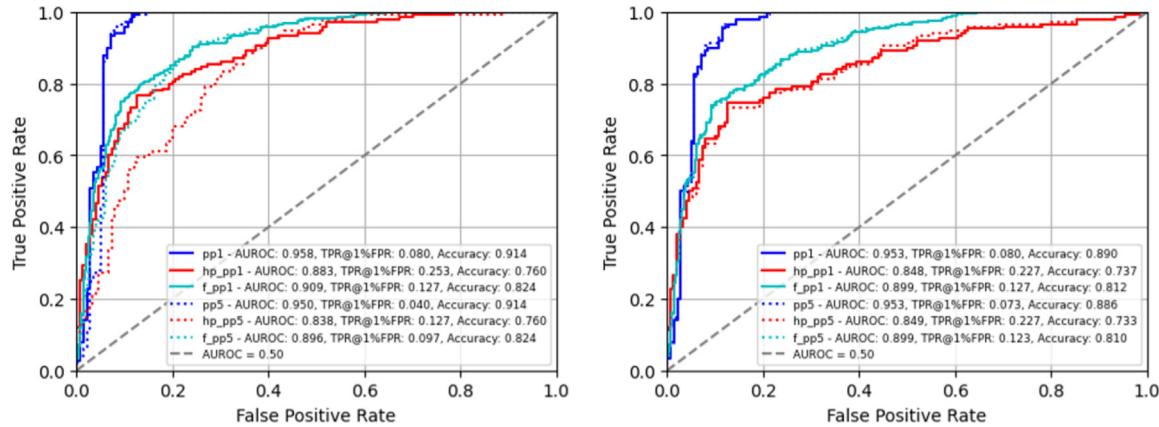


Fig. 23. ROC Curve with Full Human Data and Non-watermarked XSum LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from XSum GPT-generated Text).

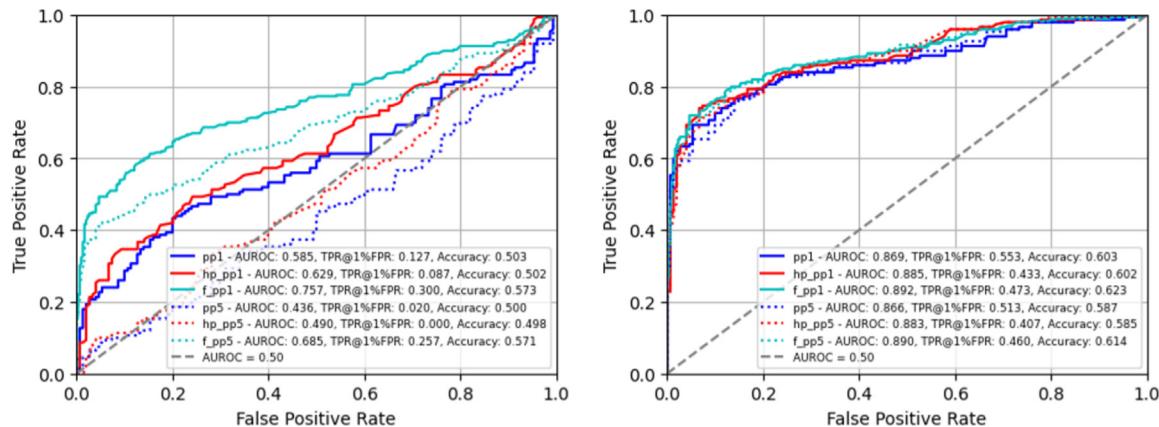


Fig. 24. ROC Curve with Full Human Data and Watermarked XSum LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from XSum GPT-generated Text).

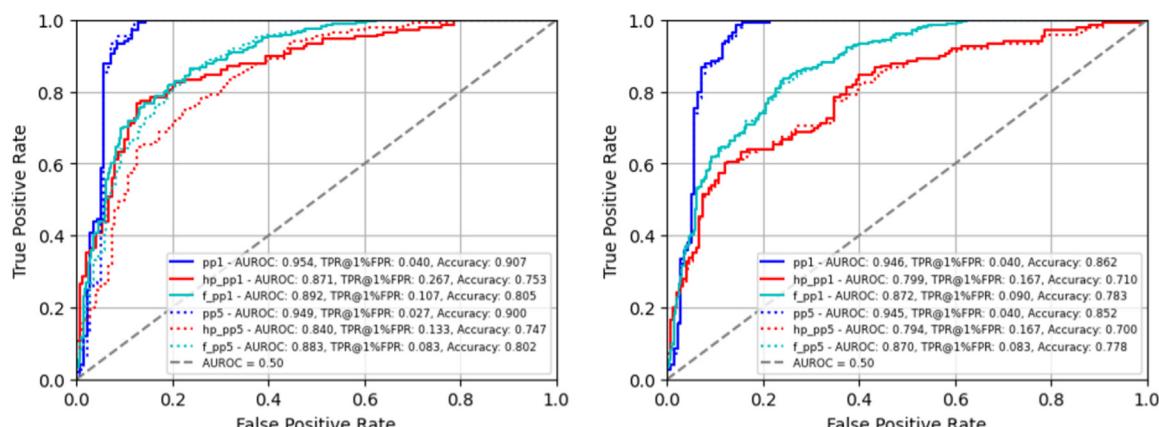


Fig. 25. ROC Curve with Full Human Data and Non-watermarked XSum LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from XSum OPT-generated Text).

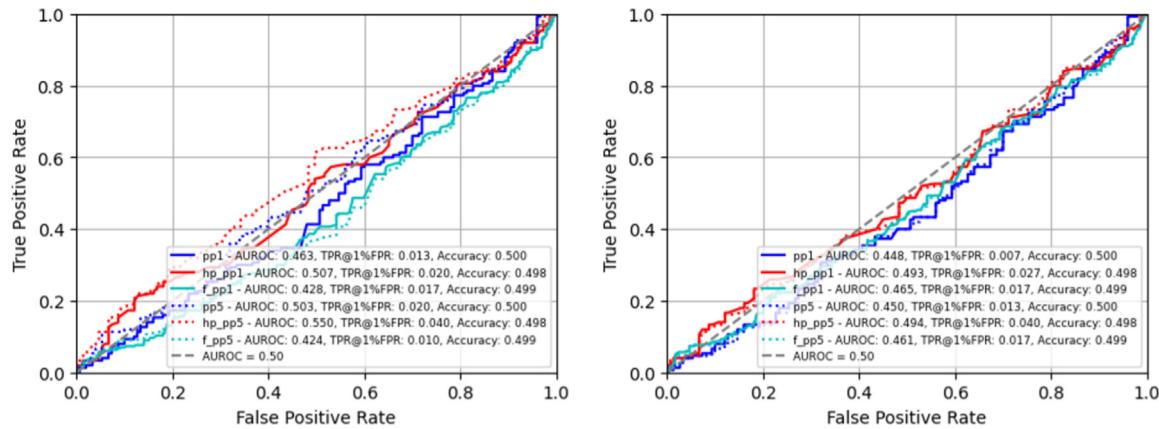


Fig. 26. ROC Curve with Full Human Data and Watermarked XSum LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from XSum OPT-generated Text).

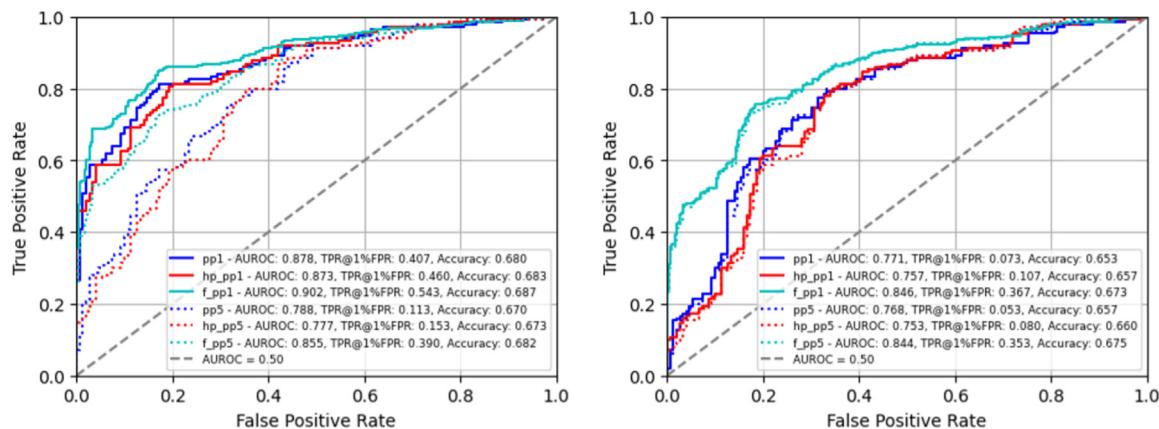


Fig. 27. ROC Curve with Full Human Data and Non-watermarked QQP LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from QQP GPT-generated Text).

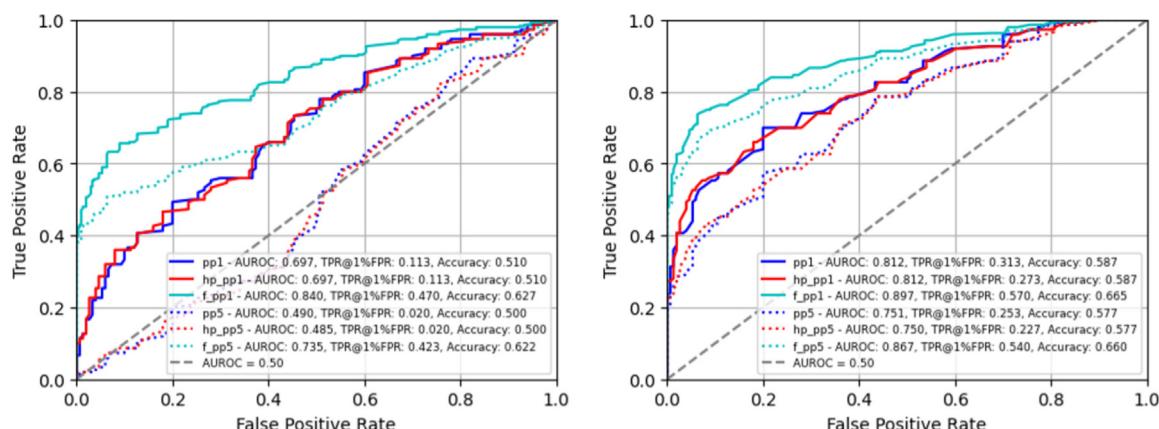


Fig. 28. ROC Curve with Full Human Data and Watermarked QQP LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from QQP GPT-generated Text).

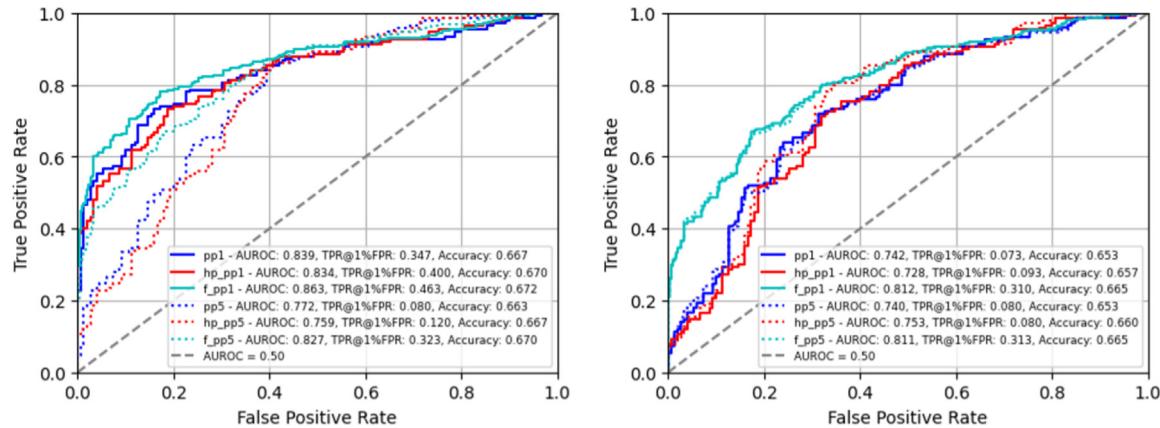


Fig. 29. ROC Curve with Full Human Data and Non-watermarked QQP LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from QQP OPT-generated Text).

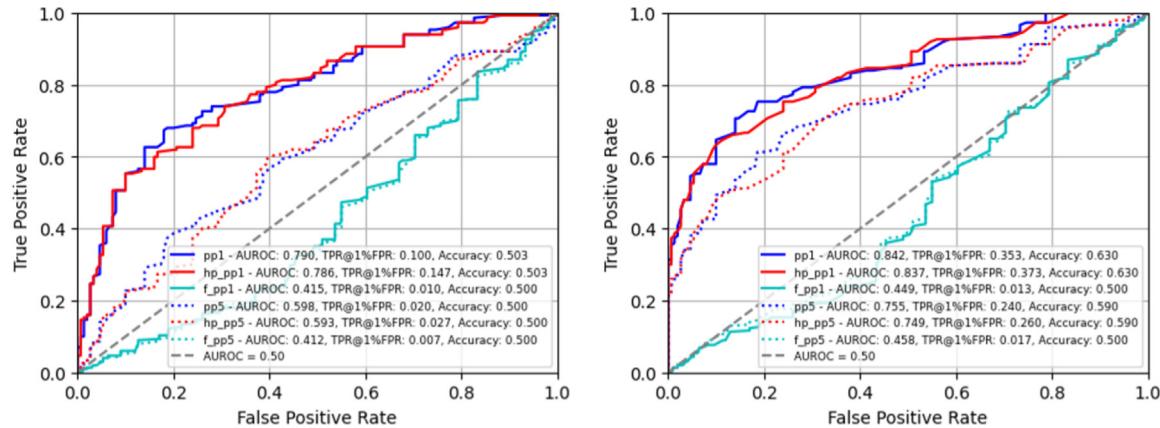


Fig. 30. ROC Curve with Full Human Data and Watermarked QQP LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from QQP OPT-generated Text).

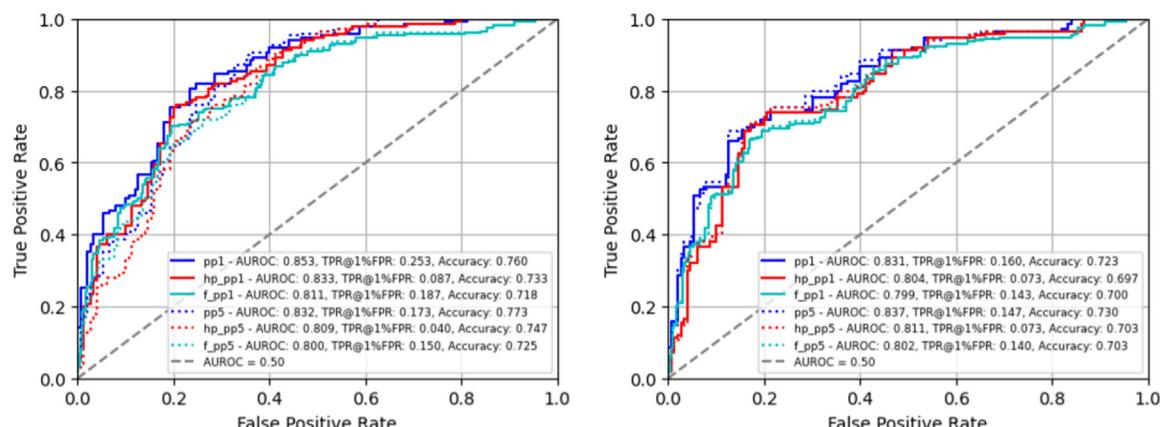


Fig. 31. ROC Curve with Full Human Data and Non-watermarked MultiPIT LLM-Data (Paraphrases generated with DIPPER (left) and MultiPIT (right) from MultiPIT GPT-generated Text).

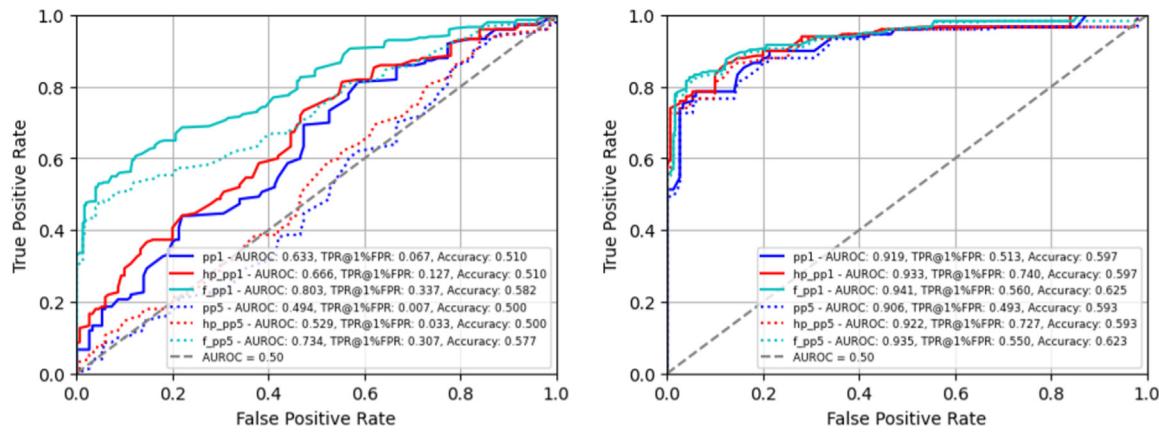


Fig. 32. ROC Curve with Full Human Data and Watermarked MultiPIT LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from MultiPIT GPT-generated Text).

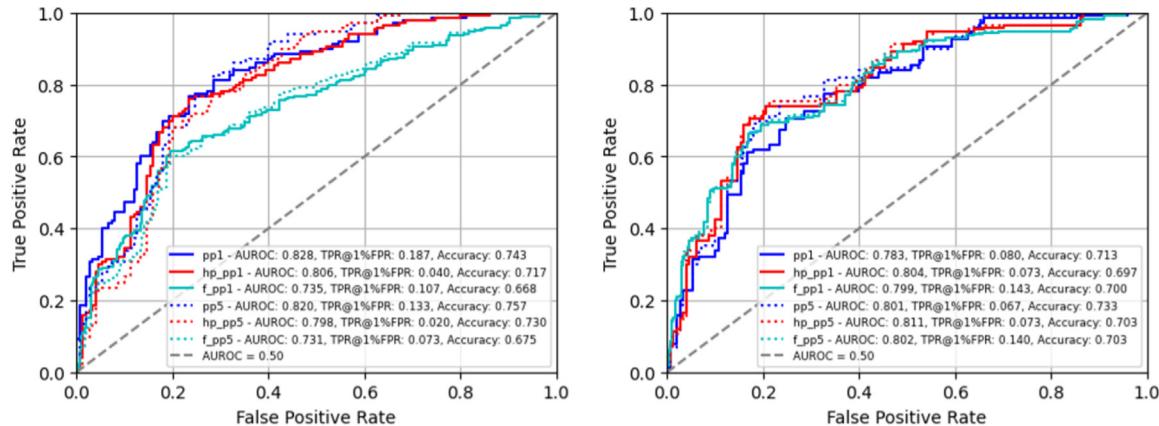


Fig. 33. ROC Curve with Full Human Data and Non-watermarked MultiPIT LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from MultiPIT OPT-generated Text).

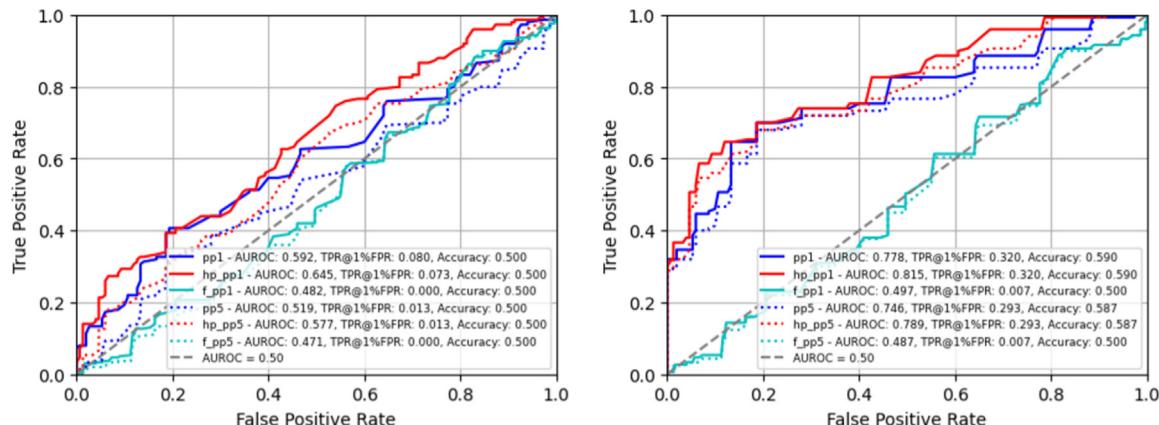


Fig. 34. ROC Curve with Full Human Data and Watermarked MultiPIT LLM-Data (Paraphrases generated with DIPPER (left) and BART (right) from MultiPIT OPT-generated Text).

References

- Allam, A.M.N., Haggag, M.H., 2012. The question answering systems: A survey. *Int. J. Res. Rev. Inf. Sci. (IJRRIS)* 2 (3).
- An, H., Acquaye, C., Wang, C., Li, Z., Rudinger, R., 2024. Do large language models discriminate in hiring decisions on the basis of race, ethnicity, and gender? ArXiv preprint arXiv:2406.10486.
- Ansari, S., 2023. Sentence_similarity_semantic_search. GitHub repository URL: https://github.com/Sakil786/sentence_similarity_semantic_search.
- Barreto, F., Moharkar, L., Shirodkar, M., Sarode, V., Gonsalves, S., Johns, A., 2023. Generative artificial intelligence: Opportunities and challenges of large language models. In: International Conference on Intelligent Computing and Networking. Springer, pp. 545–553.
- Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S., 2021. On the dangers of stochastic parrots: Can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. pp. 610–623.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* 33, 1877–1901.
- Cambridge Dictionary, 2019. PARAPHRASE | meaning in the Cambridge English Dictionary. URL: <https://www.cambridge.org/dictionary/english/paraphrase>.
- Damodaran, P., 2021. Parrot: Paraphrase generation for nlu. Parrot: Paraphrase Gener. Nlu.

- Dolan, B., Brockett, C., 2005. Automatically constructing a corpus of sentential paraphrases. In: Third International Workshop on Paraphrasing. IWP2005, URL: <https://www.microsoft.com/en-us/research/publication/automatically-constructing-a-corpus-of-sentential-paraphrases/>.
- Dou, Y., Jiang, C., Xu, W., 2022. Improving large-scale paraphrase acquisition and generation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 9301–9323.
- El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K., 2021. Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.* 165, 113679.
- Fröhling, L., Zubiaga, A., 2021. Feature-based detection of automated language models: tackling GPT-2, GPT-3 and grover. *PeerJ Comput. Sci.* 7, e443.
- Gao, T., Yao, X., Chen, D., 2021. Simcse: Simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 6894–6910.
- Goyal, R., Kumar, P., Singh, V.P., 2023. A systematic survey on automated text generation tools and techniques: application, evaluation, and challenges. *Multimedia Tools Appl.* (ISSN: 1380-7501) 82 (28), 43089–43144. <http://dx.doi.org/10.1007/s11042-023-15224-0>, URL: <https://doi.org/10.1007/s11042-023-15224-0>.
- Hutchinson, B., Prabhakaran, V., Denton, E., Webster, K., Zhong, Y., Denuy, S., 2020. Social biases in NLP models as barriers for persons with disabilities. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5491–5501.
- Ililia, L., Colleoni, E., Zyglidopoulos, S., 2023. Ethical implications of text generation in the age of artificial intelligence. *Bus. Ethics Environ. Responsib.* 32 (1), 201–210, <https://doi.org/10.1111/beer.12479> URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/beer.12479>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/beer.12479>.
- Jayawardena, L., Yapa, P., et al., 2024. Parafusion: A large-scale LLM-driven english paraphrase dataset infused with high-quality lexical and syntactic diversity. In: CS & IT Conference Proceedings, vol. 14, no. 4. CS & IT Conference Proceedings.
- Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., Goldstein, T., 2023. A watermark for large language models. In: International Conference on Machine Learning. PMLR, pp. 17061–17084.
- Kreps, S., McCain, R.M., Brundage, M., 2022. All the news that's fit to fabricate: AI-generated text as a tool of media misinformation. *J. Exp. Political Sci.* 9 (1), 104–117. <http://dx.doi.org/10.1017/XPS.2020.37>.
- Krishna, K., Song, Y., Karpinska, M., Wieting, J., Iyyer, M., 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Adv. Neural Inf. Process. Syst.* 36.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. URL: <https://arxiv.org/abs/1910.13461>, arXiv:1910.13461.
- Li, Z., Xu, X., Shen, T., Xu, C., Gu, J.-C., Tao, C., 2024. Leveraging large language models for nlg evaluation: A survey. ArXiv preprint arXiv:2401.07103.
- Lin, S., Hilton, J., Evans, O., 2022. Truthfulqa: Measuring how models mimic human falsehoods. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 3214–3252.
- Lopez, A., 2008. Statistical machine translation. *ACM Comput. Surv.* 40 (3), 1–49.
- Mishra, A.K., 2020. Abhimishra91/transformers-tutorials. URL: <https://huggingface.co/mrm8488/t5-base-finetuned-summarize-news>.
- Mitchell, E., Lee, Y., Khazatsky, A., Manning, C.D., Finn, C., 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In: International Conference on Machine Learning. PMLR, pp. 24950–24962.
- Mou, L., 2022. Search and learning for unsupervised text generation. *AI Mag.* 43 (4), 344–352. <http://dx.doi.org/10.1002/aaai.12068>, URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aaai.12068>.
- Narayan, S., Cohen, S.B., Lapata, M., 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. Association for Computational Linguistics.
- OpenAI, 2023. New and improved embedding model. URL: <https://openai.com/blog/new-and-improved-embedding-model>.
- Pew Research Center's Journalism Project, 2016. Reddit news users more likely to be male, Young and digital in their news preferences. URL: <https://www.pewresearch.org/journalism/2016/02/25/reddit-news-users-more-likely-to-be-male-young-and-digital-in-their-news-preferences/>.
- Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I., 2023. Robust speech recognition via large-scale weak supervision. In: International Conference on Machine Learning. PMLR, pp. 28492–28518.
- Razaq, A., Halim, Z., Rahman, A., Sikandar, K., 2024. Identification of paraphrased text in research articles through improved embeddings and fine-tuned BERT model. *Multimedia Tools Appl.* 83, <http://dx.doi.org/10.1007/s11042-024-18359-w>.
- Sadasivan, V.S., Kumar, A., Balasubramanian, S., Wang, W., Feizi, S., 2023. Can AI-generated text be reliably detected?. ArXiv preprint arXiv:2303.11156.
- Shewale, R., 2023. 30+ Google bard statistics 2023 (trends & demographics). URL: <https://www.demandsgage.com/google-bard-statistics/>.
- Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J.W., Kreps, S., et al., 2019. Release strategies and the social impacts of language models. ArXiv preprint arXiv:1908.09203.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahaire, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023. Llama 2: Open foundation and fine-tuned chat models. ArXiv preprint arXiv:2307.09288.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R., 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. URL: <https://arxiv.org/abs/1804.07461>, arXiv:1804.07461.
- Wolff, M., Wolff, S., 2020. Attacking neural text detectors. ArXiv preprint arXiv: 2002.11768.
- Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., Choi, Y., 2019. Defending against neural fake news. *Adv. Neural Inf. Process. Syst.* 32.
- Zhang, B., Haddow, B., Birch, A., 2023. Prompting large language model for machine translation: A case study. In: International Conference on Machine Learning. PMLR, pp. 41092–41110.
- Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., Hashimoto, T.B., 2024. Benchmarking large language models for news summarization. *Trans. Assoc. Comput. Linguist.* 12, 39–57.
- Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al., 2022. Opt: Open pre-trained transformer language models. ArXiv preprint arXiv:2205.01068.
- Zhao, J., Chen, K., Yuan, X., Qi, Y., Zhang, W., Yu, N., 2024. Silent guardian: Protecting text from malicious exploitation by large language models. *IEEE Trans. Inf. Forensics Secur.*.
- Zubiaga, A., 2024. Natural language processing in the era of large language models. *Front. Artif. Intell.* 6, 1350306.