MFD: Multi-Feature Detection of LLM-Generated Text

Zhendong Wu (wzd@hdu.edu.cn)
Hangzhou Dianzi University
Hui Xiang
Hangzhou Dianzi University

Article

Keywords:

Posted Date: August 9th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3226684/v1

Additional Declarations: No competing interests reported.

MFD: Multi-Feature Detection of LLM-Generated Text

Hui Xiang[1] **and Zhendong Wu**[1,*]

[1]Hangzhou Dianzi University, Hangzhou, China
[*]wzd@hdu.edu.cn

ABSTRACT

With the rapid development of large language models, their powerful capabilities have led to their rapid popularity in society. However, it not only brings great convenience to people's life and work but also provides a favorable tool for criminals to carry out malicious behaviors. Therefore, to prevent the malicious use of large language models, there is a growing demand for a detector that can efficiently discriminate texts generated by large language models. In this paper, Multi-Feature Detection (MFD), a new zero-shot method, is introduced. MFD comprehensively considers log-likelihood, log-rank, entropy, and LLM-Deviation. LLM-Deviation is a new statistical feature proposed in this paper and has a clear distribution difference between texts generated by LLMs and those written by humans. Experiments show MFD is more effective than the existing zero-shot method. MFD improves the detection performance by 1.02 F1 score on average on the HC3-English dataset. In generalization ability, MFD is also very competitive compared with the existing zero-shot method.

Introduction

In recent years, large language models (LLMs) are developing rapidly. LLMs such as ChatGPT[1], LLaMa[2], and BLOOM[3] have shown powerful capabilities in dialogue generating, question answering, information retrieval, content continuation, literature authoring, etc. They also can generate code, debug code, and generate comments for code. The recently released GPT-4[4] not only achieves further improvements in performance on various tasks of natural language processing but also is a multi-modal model. The emergence of large language models has had an impact and influence on human society.

However, misuse of LLMs can lead to many negative consequences. Firstly, LLMs occasionally exhibit various undesirable behaviors. For instance, LLMs may copy harmful or biased content or produce so-called hallucinating outputs by fabricating non-existent or false facts. Secondly, users may use it for unethical purposes. For example, students can use LLMs to complete papers and assignments, and malicious actors can use LLMs to generate spam, fake news, and fake reviews[5]. So, the misuse of LLMs is harmful to education and society[6].

To solve the above problems, previous researchers designed many detection methods to discriminate between LLM-generated texts and human-written texts. The existing detection methods can fall into three categories: the PLM-based methods, the feature-based methods, and the LLM-based methods. PLM-based methods[6–10] mainly distinguish LLM-generated texts from human-written texts by fine-tuning the encoder-only pre-trained models (PLMs), such as BERT[11], RoBERTa[12], etc. Feature-based methods[6, 13–16] distinguish based on the distribution difference of a statistical feature between LLM-generated text and human-written text, such as perplexity, log-likelihood, rank, entropy, negative curvature, and so on. LLM-based methods[17–20] can distinguish by watermarking the text generated by LLMs or recording all the text generated by LLMs for retrieval.

Each of these three category of methods has its advantages and disadvantages. Although PLM-based methods can achieve good performance, fine-tuning the pre-trained model to ensure its high performance is often necessary for each new LLM or new dataset, which is expensive. LLM-based methods are only feasible for LLM developers, not for third parties. Feature-based methods, also known as zero-shot methods, require no additional fine-tuning of the pre-trained model and support third-party implementations. However, the statistical features are calculated based on the probability distribution of tokens. The LLM that can access the probability distribution of tokens names the white-box model, otherwise the black-box model. Thus feature-based methods are mainly applicable to the white-box model. While for the black-box model, a conventional approach is using the white-box model as a proxy model.

Therefore, this paper mainly considers methods based on statistical features and introduces a new zero-shot method called Multi-Feature Detection (MFD). MFD comprehensively considers log-likelihood, log-rank, entropy, and LLM-Deviation, and uses the neural network model as the classification model. Experiments show MFD effectively improves the detection performance. LLM-Deviation is a new statistical feature introduced in this paper, which can measure the difference between the detected text and the text generated by the LLMs in the ideal state. There is a clear distribution difference in LLM-Deviation between LLM-generated texts and human-written texts. About experiments, the in-domain and out-of-domain tests are performed on five datasets (finance, medicine, open_qa, reddit_eli5, and wiki_csai) contained in HC3-English[7]. In addition, out-of-domain tests are also conducted on three datasets (TruthfulQA, SQuAD1, NarrativeQA) used by He et al. (2023)[8]. Finally, to further analyze several factors that determine the detection performance of MFD, some ablation studies are carried out. All experiments are performed on MGTBench[8].

In summary, our contributions are as follows:

- Multi-Feature Detection (MFD), a new zero-shot method, is introduced. MFD is more accurate in detecting LLM-generated text than SOTA in the existing zero-shot method. On the HC3-English dataset, the detection performance is increased by 1.02 F1 score on average.

- LLM-Deviation, a new statistical feature, is proposed. LLM-Deviation has a clear distribution difference between texts generated by LLMs and texts written by humans. LLM-generated texts typically have smaller LLM-Deviation than human-written texts.

- Evaluating the generalization ability of MFD, the result shows MFD's generalization ability is also competitive with existing zero-shot methods.

- What factors can determine the detection performance of MFD is analyzed through ablation studies.

Related Work

The LLM-generated text detection problem is usually regarded as a binary classification problem. The existing detection methods can fall into three categories: the PLM-based methods, the feature-based methods, and the LLM-based methods.

PLM-based methods fine-tune the pre-trained model through supervised learning to detect the text generated by LLMs. The training dataset consists of LLM-generated text and human-written text. Solaiman et al. (2019)[6] developed a GPT-2 Detector by fine-tuning RoBERTa to detect text generated by GPT-2. The training set consists of texts generated by GPT-2 and those written by humans. Guo et al. (2023)[7] built the HC3 dataset, which contains text generated by ChatGPT and text written by humans, and used this dataset to fine-tune RoBERTa to develop the ChatGPT Detector. He et al. (2023)[8] developed an LM Detector obtained by fine-tuning BERT. The dataset used for fine-tuning contains three question-answering datasets, each containing generated text from six LLMs and text written by humans. According to the different characteristics of long and short texts, Tian et al. (2023)[9] proposed a Multiscale Positive-Unlabeled (MPU) training framework based on Positive-Unlabeled (PU) learning. This framework mainly fine-tunes the pre-trained model by transforming the detection problem of LLM-generated text into a partial PU learning problem. Antoun et al. (2023)[10] fine-tuned the XLM-R[21] (a multi-lingual RoBERTa model) with the translated HC3-English dataset to develop a multi-language ChatGPT detector. The detector with fine-tuning of the pre-trained model has good performance. However, Bakhtin et al. (2019)[22]; Uchendu et al. (2020)[23] pointed out that detectors obtained by fine-tuning pre-trained models tend to overfit their in-domain datasets or source models. Therefore, fine-tuning a new detector is usually indispensable for achieving high detection performance in each new dataset or LLM.

Feature-based methods mainly discriminate between LLM-generated and human-written texts by the distribution difference of statistical features. Firstly, the text is input into LLM to obtain the corresponding statistical features. Then these statistical features are used to train a detector. Gehrmann et al. (2019)[14] proposed the statistical features can be the probability of each word in the text, the absolute rank of the probability of each word, or the entropy of the predicted distribution of words at each position, and developed a visualization tool for the probability distribution, rank distribution and entropy distribution of text words to help humans detect the text generated by LLMs. Mitchell et al. (2023)[15] proposed a hypothesis: the log-likelihood of text generated by LLMs after a small amount of rewriting will tend to be lower than the original text, while human-written text can be higher or lower than the original text. And DetectorGPT was built based on this assumption. Su et al. (2023)[16] proposed two new zero-shot methods. One, DetectLLM-LRR, uses the Log-Likelihood Log-Rank Ratio as a statistical feature. The other one, DetectLLM-NPR, is based on DetectGPT and uses log-rank instead of log-likelihood.

LLM-based methods must access the model architecture of LLMs, so only LLM developers can implement it, and it is not feasible for third parties. Abdelnabi and Fritz (2021)[17] proposed the Adversarial Watermarking Transformer (AWT), which is an end-to-end model that encodes information by learning word substitutions and their positions, thus hiding watermarks in the text. Grinbaum and Adomaitis (2022)[18] proposed to use a code based on equidistant letter sequences to watermark the text generated by LLMs. Kirchenbauer et al. (2023)[19] proposed to generate watermarked text by using "green" tokens as much as possible during the sampling process. These "green" tokens are randomly selected before LLMs generate each word. In addition to watermarking techniques, Krishna et al. (2023)[20] proposed to have the LLMs record each generated text and store it in a database. If there is a text in the database whose semantic similarity with the detected text exceeds the threshold, it means the detected text is the text generated by the LLMs.

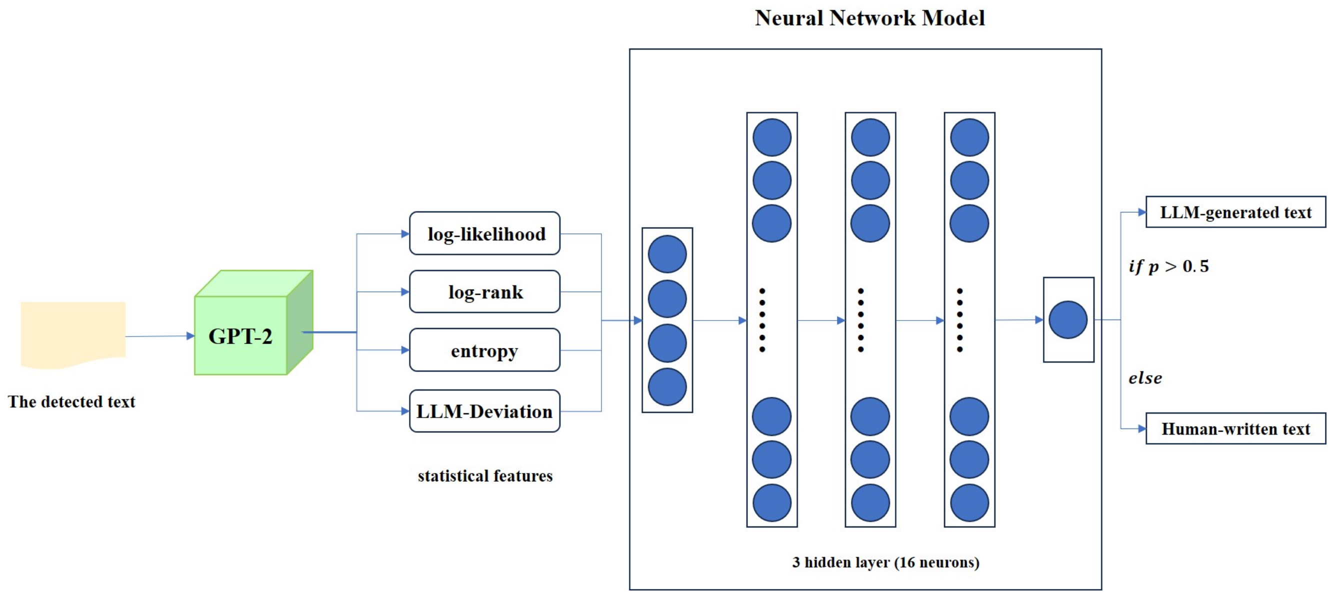

Figure 1. The overview of MFD.

A Robust LLM-generated Text Detection Method – Multi-Feature Detection

In general, every LLM designed and trained by the developer has a certain style of text. These styles can be learned by text statistical feature analysis, and then recognized by lightweight detection models. Previous zero-shot methods usually consider only one statistical feature, such as perplexity, probability, rank, and entropy[6,14,15]. Even for the latest zero-shot method, DetectLLM-LRR[16], researchers define the connection between probability and rank by themselves and integrate these two statistical features into one statistical feature. Therefore, the previous zero-shot methods are vulnerable to statistical feature distribution shifts. Given the above phenomenon, inspired by ensemble learning, this paper proposes multi-feature detection. Multi-feature detection can be more robust than single-feature detection because even if one of the statistical features has a distribution shift problem, the other statistical features can compensate for the problem. At the same time, there must be connections among various statistical features, which makes multi-feature detection have better detection performance than single-feature detection. The connections are not easily discoverable and represented, but neural network models can learn deeper connections between statistical features, which are relatively stable and not easily disturbed. In summary, the multi-feature detector obtained by learning the deep statistical features of the text can take more accurate and robust LLM-generated text detection performance. Based on the above motivation, Multi-Feature Detection (MFD), a new zero-shot method, is introduced in this section. And the specific design of MFD is as follows.

Multi-Feature Detection

Specifically, MFD can fall into two modules: input features and classification model.

Regarding input features, the features should show differences between LLM-generated and human-written texts. Previous studies have found distributional differences in log-likelihood, log-rank, and entropy between texts generated by LLMs and those written by humans. LLM-generated texts generally have higher log-likelihood, smaller log-rank, and lower entropy than human-written texts. Therefore, MFD uses log-likelihood, log-rank, entropy, and LLM-Deviation as input features. LLM-Deviation is a new statistical feature proposed in this paper and described in the LLM-Deviation subsection. The log-likelihood, log-rank, and entropy follow the setting of previous works[6,14,15]. The log-likelihood is defined as

$$log\ likelihood = \frac{1}{t}\sum_{i=1}^{t} \log p_\theta(x_i|x_{<i}), \tag{1}$$

the log-rank is defined as

$$log\ rank = \frac{1}{t}\sum_{i=1}^{t} \log r_\theta(x_i|x_{<i}), \tag{2}$$

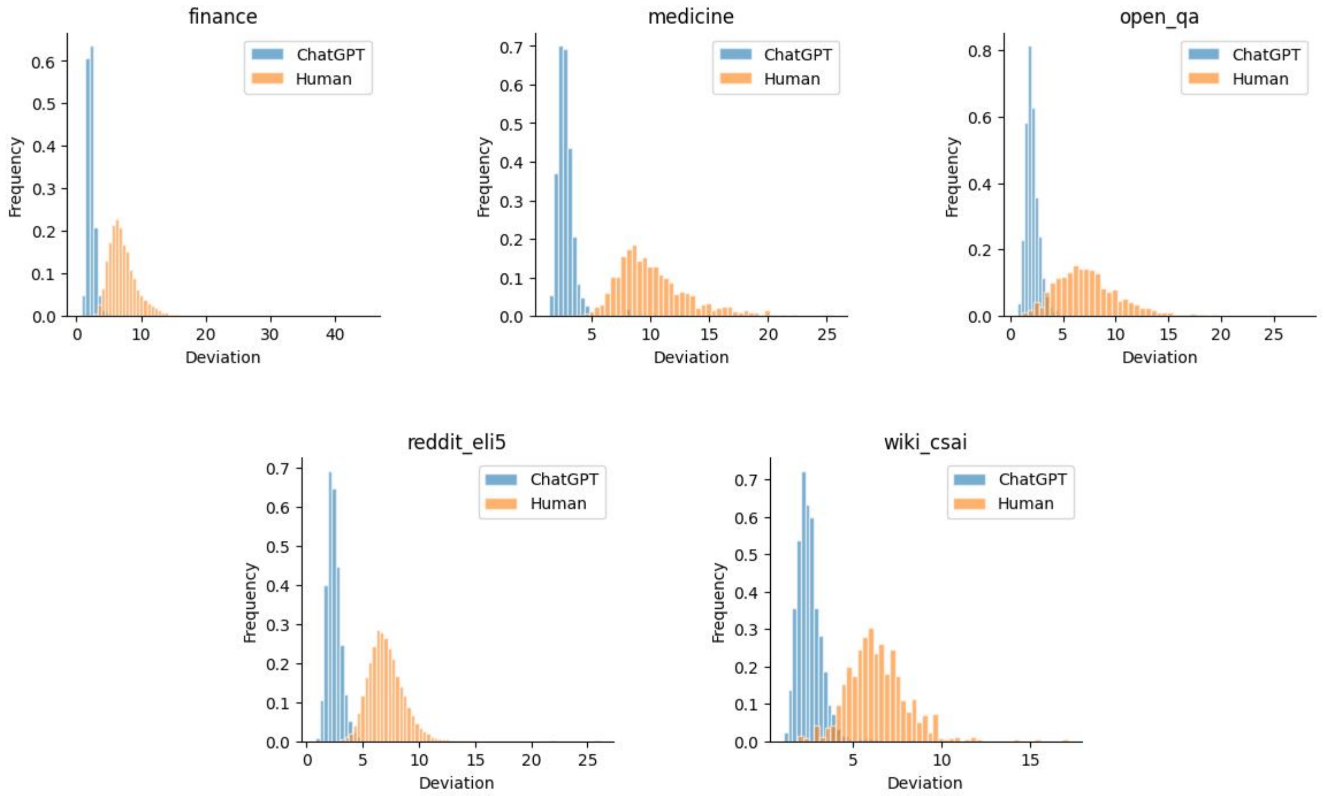

Figure 2. LLM-Deviation distribution plots of texts of five datasets in HC3-English.

the entropy is defined as

$$entropy = -\frac{1}{t}\sum_{i=1}^{t}\sum_{w} p_\theta(x_i = w|x_{<i})\log p_\theta(x_i = w|x_{<i}), \tag{3}$$

where $p_\theta(x_i|x_{<i})$ denotes the probability of token x_i, $r_\theta(x_i|x_{<i}) \geq 1$ denotes the absolute rank of probability of token x_i, both are calculated by LLM according tokens before position i of text $X = (x_1, x_2, \ldots, x_t)$. Since the source of the detected text is uncertain, and many LLMs are black-box models, a white-box model is usually used as the proxy model to compute the statistical features of the detected text. While among all white-box models, GPT-2 is not very large in scale and can be deployed on the local computer. Therefore, MFD uses GPT-2 as the proxy model to calculate the four statistical features.

Regarding the classification model, the neural network model is selected as the classification model because neural network models can find and describe the connections among statistical features better than machine learning classification models. The larger the neural network model scale, the more expensive it is to train. In the case of a small number of input features, there is no need to design too deep a neural network model. Therefore, the number of neurons in the hidden layer is limited to 4,8,16, and the number of layers of the hidden layer is limited in the range between 1 and 4. After conducting a series of experiments, MFD's classification model uses a 5-layer neural network model for trading off the performance and time of the detector. The neural network model consists of one input layer, three hidden layers, and one output layer. The input layer is composed of 4 neurons. Each hidden layer is composed of 16 neurons and ReLU activation functions. And the final output layer is composed of 1 neuron and Sigmoid activation functions.

To sum up, MFD uses log-likelihood, log-rank, entropy, and LLM-Deviation as input features and uses a neural network as the classification model. MFD first uses GPT-2 as a proxy model to calculate four statistical features of the detected texts (log-likelihood, log-rank, entropy, and LLM-Deviation). Then the four statistical features are input into the neural network model and go through three hidden layers. Finally, the output layer outputs the probability that the detected text is text generated by LLMs. The detected text is LLM-generated if the output probability is above 0.5, else human-written. See Figure 1 for an overview of MFD. In the Experiment section, the results show MFD is a better detection method than the existing zero-shot methods.

LLM-Deviation

LLM-Deviation is defined as

$$LLM\ Deviation = \frac{1}{t}\sum_{i=1}^{t}(\log r_\theta(x_i|x_{<i}))^2, \tag{4}$$

where $r_\theta(x_i|x_{<i}) \geq 1$ denotes the absolute rank of probability of token x_i, which calculated by LLM according tokens before position i of text $X = (x_1, x_2, \dots, x_t)$.

The output tokens of LLMs in the ideal case all have the smallest rank, then the average log-rank of the text is 0. Therefore, LLM-Deviation represents the variance of the log-rank of the detected text tokens in case enforcing zero as the mean, which can evaluate the deviation of the detected text and the text generated by the LLMs in the ideal state. In general, LLM-Deviation of the LLM-generated texts is smaller than the human-written texts, as shown in Figure 2. Thus, LLM-Deviation can be used as a statistical feature to distinguish texts generated by LLMs from texts written by humans.

Experiment

In this section, the experimental setting is described first. Then, the evaluation results and ablation studies are analyzed.

Datasets & Metrics

The experimental dataset is HC3-English from the Human ChatGPT Comparison Corpus (HC3)[7]. HC3-English contains five datasets: finance, medicine, open_qa, reddit_eli5, and wiki_csai. Each dataset consists of questions, human responses, and ChatGPT responses. When performing in-domain tests on each dataset, the training set consists of 80% of the dataset, and the test set consists of the remaining 20%. While performing out-of-domain tests on each dataset, the training set consists of 80% of the dataset, and the test set consists of 20% of every dataset in other four datasets. In addition, out-of-domain tests are also conducted on the three datasets used by He et al. (2023)[8], namely TruthfulQA, SQuAD1, and NarrativeQA.

Following previous works (Gehrmann et al., 2019[14]; Mitchell et al., 2023[15]), the metric to measure the performance of the detector is the F1 score. A greater F1 score indicates a better performance of the detector.

Zero-Shot Methods

The following zero-shot methods are used as baseline models:

Log Likelihood: First, this method uses an LLM to calculate the probability of each token x_i in the text $X = (x_1, x_2, \dots, x_t)$, then uses the average log-probability of each token in the text as a statistical feature. This value is larger, and the probability that the detected text is LLM-generated text is higher.

Log Rank: According to the probability distribution of each token in detected text $X = (x_1, x_2, \dots, x_t)$, the absolute rank of each token can be calculated. This method uses the average log-rank of each token in the detected text as a statistical feature. This value is smaller, and the probability that the detected text is LLM-generated text is higher.

Entropy: The average entropy of the probability distribution of each token in the detected text is used as the statistical feature. The smaller this value is, the higher the probability.

GLTR: The rank of each token in the detected text is first obtained. Then all tokens in the detected text are divided into four groups according to their rank with 10, 100, and 1000 as boundaries. Finally, the percentage of the number of tokens contained in these four groups is taken as a set of statistical features. This set of statistical features is used as input to a classifier, which outputs the probability that the detected text is a text generated by LLMs.

DetectGPT: Firstly, the perturbation model generates several perturbed texts of the detected text. The average difference between the log probabilities of the original text and several perturbed texts is then measured and perceived as a statistical feature.

DetectLLM-LRR: In this method, the statistical feature is the Log-Likelihood Log-Rank Ratio of the detected text.

Experimental Details

Following previous works, the binary classification model is the logistic regression model. However, due to resource constraints, GPT2-medium is selected as the LLM for computing the probability distribution for each token in the text, and the maximum token length of the text is limited to 512. For DetectGPT, the perturbation model is T5-large[24], and the maximum number of perturbed texts is 10. In addition, we find the two detectors built by Su et al. (2023)[16], DetectLLM-LRR and DetectLLM-NPR, have the problem of zero denominator, so add a smoothing term $\varepsilon = 10^{-8}$ to the denominator when using Detect-LRR as the baseline model.

Table 1. In-domain tests. Comparison of MFD to other zero-shot method baselines in terms of F1 score. Bold indicates the best results between different methods.

Method	finance	medicine	open_qa	reddit_eli5	wiki_csai	average
Log Likelihood	97.59	99.40	94.09	98.53	95.91	97.10
Log Rank	97.34	99.40	95.28	98.44	96.49	97.39
Entropy	93.62	96.59	88.57	93.14	79.27	90.24
GLTR	97.47	99.00	94.97	98.34	95.98	97.15
DetectGPT	83.92	90.24	57.14	89.20	87.03	81.51
DetectLLM-LRR	96.69	98.58	95.44	97.82	95.58	96.82
MFD(ours)	**98.60**	**99.60**	**97.07**	**99.40**	**97.39**	**98.41**

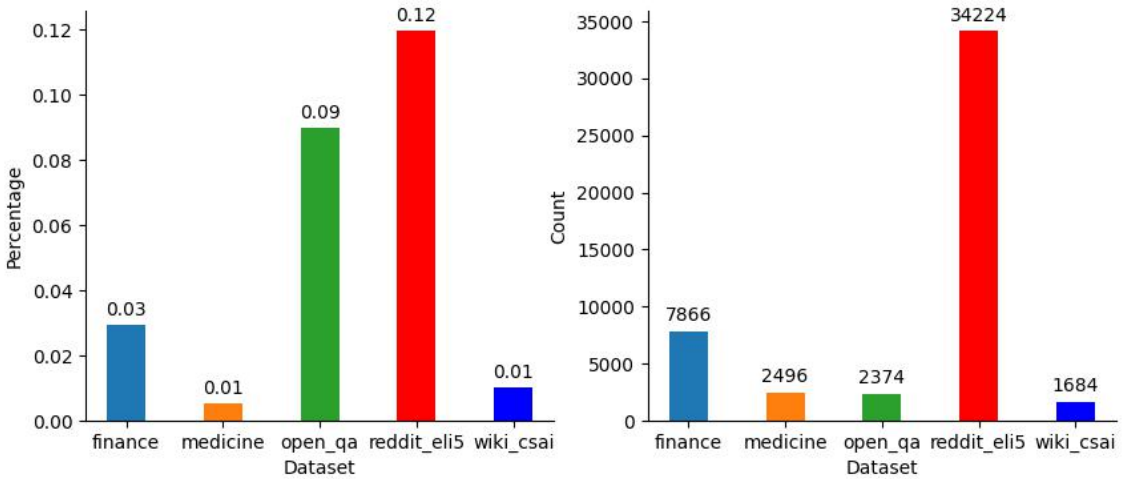

Figure 3. The left plot represents the proportion of texts in the dataset that are longer than 512. The right plot represents the number of texts in the dataset.

Evaluation Results

In-Domain

Table 1 shows the comparison results of MFD and the six zero-shot methods introduced above on each of the five in-domain datasets. It can be seen MFD achieves the best performance on all five datasets. The average performance of MFD is 1.02 F1 score higher than the SOTA of the existing zero-shot methods. In addition, all zero-shot methods achieve the best performance on the medicine dataset. And, except for entropy, the other methods achieve the worst performance on the open_qa dataset. Through analyzing the datasets, the result shown in Figure 3, we guess it is because the proportion of texts that are more than 512 tokens in the medicine dataset is the smallest among datasets of similar size, while open_qa is the largest. Moreover, although the proportion of texts that are more than 512 tokens in the reddit_eli5 dataset is the largest among all datasets, due to the largest dataset size leads to the largest amount of data used for training, which makes up for this problem to a certain extent. Thus almost all zero-shot methods achieve the second-best performance on the reddit_eli5.

Out-of-Domain

To evaluate the generalization ability of MFD, MFD compares with the above six zero-shot methods on out-of-domain datasets. The results are shown in Table 2. The performance of MFD on out-of-domain datasets is very competitive compared with the existing zero-shot methods. Whether the training set is finance or medicine, MFD obtains the best average performance on the out-of-domain datasets. In addition, detectors trained on different datasets may have different transferability to other datasets. This finding suggests the choice of the dataset used to train influences the detector generalization ability. An interesting finding is that detector trained on the B dataset may has better performance on the A dataset than the detector trained on the A dataset. For example, the Entropy detector trained in finance, reddit_eli5, or wiki_csai dataset has better detection performance on the medicine test set than the Entropy detector trained in the medicine dataset. The specific reasons can be further explored.

To further investigate the generalization ability of the zero-shot methods, all the zero-shot methods are first trained on the

Table 2. Out-of-domain tests. Comparison of MFD to other zero-shot method baselines in terms of F1 score. Bold indicates the best results between different methods.

Train	Test	Log Likelihood	Log Rank	Entropy	GLTR	DetectGPT	DetectLLM-LRR	MFD(ours)
finance	medicine	**99.19**	98.99	96.93	98.17	90.50	93.59	98.59
	open_qa	90.29	91.51	86.88	89.77	55.53	**95.53**	91.12
	reddit_eli5	98.65	98.59	88.65	98.26	89.81	97.70	**99.06**
	wiki_csai	94.32	95.40	79.55	94.62	83.46	95.55	**95.70**
	average	95.61	**96.12**	88.00	95.21	79.83	95.59	**96.12**
medicine	finance	95.67	94.61	92.50	94.51	84.41	92.51	**97.49**
	open_qa	86.50	87.13	85.71	87.13	55.98	**91.33**	88.27
	reddit_eli5	97.91	96.79	92.08	97.30	89.54	94.30	**98.65**
	wiki_csai	90.57	90.08	80.31	89.60	84.60	93.33	**94.35**
	average	92.66	92.15	87.65	92.13	78.63	92.87	**94.69**
open_qa	finance	95.91	96.93	87.48	**97.95**	82.23	94.74	95.63
	medicine	92.21	92.21	85.25	**97.53**	88.55	81.71	90.07
	reddit_eli5	89.38	93.28	63.84	**96.66**	88.03	93.91	87.69
	wiki_csai	92.83	93.83	61.78	**96.12**	80.49	91.77	90.51
	average	92.58	94.06	74.59	**97.06**	84.83	90.53	90.97
reddit_eli5	finance	97.05	97.10	91.65	97.05	83.56	97.03	**98.04**
	medicine	**99.60**	99.39	96.84	98.79	88.43	94.94	97.61
	open_qa	89.27	90.63	83.75	89.27	51.49	**95.55**	83.24
	wiki_csai	93.58	94.59	79.01	94.08	83.64	95.60	**96.83**
	average	94.87	95.43	87.81	94.80	76.78	**95.78**	93.93
wiki_csai	finance	**97.94**	97.83	93.59	97.71	74.05	96.70	97.72
	medicine	97.95	97.74	97.13	**98.16**	80.48	93.82	97.14
	open_qa	92.16	93.68	87.64	91.68	40.38	**95.33**	88.45
	reddit_eli5	97.15	98.26	85.97	97.99	81.37	97.78	**98.62**
	average	96.30	**96.88**	91.08	96.39	69.06	95.91	95.48

Table 3. Out-of-domain tests. All zero-shot methods are trained on finance and then transferred to TruthfulQA, SQuAD1, and NarrativeQA. The evaluation metric is the F1 score.

Test	Log Likelihood	Log Rank	Entropy	GLTR	DetectGPT	DetectLLM-LRR	MFD(ours)
TruthfulQA	80.00	77.74	50.69	79.58	76.12	66.93	82.28
SQuAD1	6.76	5.48	7.95	13.41	20.78	5.19	6.63
NarrativeQA	3.80	3.80	0	4.88	26.82	1.12	9.38

finance dataset and then transferred to the three datasets used by He et al. (2023)[8]: TruthfulQA, SQuAD1, and NarrativeQA. Table 3 shows all zero-shot methods perform poorly on SQuAD1 and NarrativeQA. This finding suggests none of these zero-shot methods are truly universal detectors. We believe the inherent distribution differences among different types of datasets, the differences that may exist in text generation rules among different LLMs, and the differences in additional conditions when the same LLM generates text make developing a general detector very difficult.

Ablation Studies

To explore the components that affect the performance of MFD, the possible effect caused by different choices of the classification model is analyzed first. Then, the possible effect caused by the composition of input features of the classification model is analyzed.

Deep Learning or Machine Learning

By evaluating the detection performances of the two different MFDs, the effect that may result from different choices of the classification model is analyzed. One uses the logistic regression model from machine learning as the classification model, called MFD, and the another uses the neural network model from deep learning, called MFD-M. Table 4 shows the MFD performed the best on all five datasets. The average performance of MFD on five datasets is 0.65 F1 score better than MFD-M. Because the logistic regression model represents a linear relationship, while the neural network model represents a nonlinear

Table 4. The F1 score of detection performance of detectors using two different classification models. Bold indicates the best results between different methods.

Method	finance	medicine	open_qa	reddit_eli5	wiki_csai	average
MFD-M	98.10	99.20	95.45	98.95	97.09	97.76
MFD	**98.60**	**99.60**	**97.07**	**99.40**	**97.39**	**98.41**

Table 5. Average F1 score on the HC3-English dataset. The best performance is indicated in bold.

Feature				HC3-English average F1 score
log-likelihood	log-rank	entropy	LLM-Deviation	
✓	✗	✗	✗	96.86
✗	✓	✗	✗	97.15
✗	✗	✓	✗	90.42
✓	✓	✗	✗	97.47
✓	✗	✓	✗	98.04
✗	✓	✓	✗	98.06
✓	✓	✓	✗	98.37
✓	✓	✓	✓	**98.41**

relationship, the relationships among the statistical features log-likelihood, log-rank, entropy, and LLM-Deviation are more likely nonlinear. And the use of a neural network model can better capture the nonlinear characteristics among them.

Features

By removing part input features of MFD, then measuring the average performance of MFD in the five datasets, whether three input features of MFD (log-likelihood, log-rank, and entropy) are all necessary can be analyzed. Table 5 shows the results. Removing either of the three statistical features reduces the detection performance of MFD. And the performance of removing any two statistical features is worse than the performance of removing any one statistical feature. Therefore, these several statistical features can complement each other. And the detection performance of MFD can achieve the best only when these several statistical features are all present. In addition, removing entropy has a prominent impact on the detection performance of MFD. The log-likelihood and entropy ensemble and the log-rank and entropy ensemble improve performance better than the log-likelihood and log-rank ensemble. Because log-rank is essentially the same as log-likelihood, both are based on a general rule of LLMs to generate text–select tokens with high probability as outputs, while entropy is according to the entire probability distribution. But there is still a difference between log-likelihood and log-rank. The log-likelihood is based on probability, while the log-rank on rank. Even for the same rank, it is possible to have different probabilities. Similarly, with the same probability, it is possible to have different ranks. So the composition of log-likelihood and log-rank is also beneficial to improve performance.

Then, to analyze whether LLM-Deviation, the custom statistical feature, plays a role in MFD, MFD with and without LLM-Deviation are evaluated on the HC3-English dataset in the average performances. The last two columns of Table 5 show that LLM-Deviation is effective in improving the performance of MFD. The average detection performance of MFD with LLM-Deviation as an input feature in five datasets is 0.04 F1 score better than the MFD without LLM-Deviation as an input feature.

Conclusion

In this paper, Multi-Feature Detection (MFD), a new zero-shot method, is introduced. MFD takes four statistical features, log-likelihood, log-rank, entropy, and LLM-Deviation, into account and uses the neural network model as the classification model. Among these statistical features, LLM-Deviation is a new statistical feature proposed in this paper, which has a clear distribution difference between texts generated by LLMs and those written by humans. Texts generated by LLMs often have smaller LLM-Deviation than texts written by humans. Experiments show MFD achieves state-of-the-art performance on the HC3-English dataset. In generalization ability, MFD is also competitive compared with existing zero-shot methods. Moreover, the relationships among log-likelihood, log-rank, entropy, and LLM-Deviation are more likely nonlinear. And these statistical features complement each other and are all necessary for MFD to perform at its best.

Limitations

Due to the inherent distribution differences among different types of datasets, the differences that may exist in text generation rules among different LLMs, and the differences in additional conditions when the same LLM generates text, developing a general detector is extremely difficult. However, training a new detector for each case is expensive. Therefore, in the future, the generalization ability of the detector is an issue to be further solved.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

1. OpenAI. Chatgpt: Optimizing language models for dialogue. https://openai.casa/blog/chatgpt/ (2022).

2. Touvron, H. *et al.* Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

3. Scao, T. L. *et al.* Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100* (2022).

4. OpenAI. Gpt-4 technical report. *ArXiv* **abs/2303.08774** (2023).

5. Adelani, D. I. *et al.* Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In *Advanced Information Networking and Applications: Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020)*, 1341–1354 (Springer, 2020).

6. Solaiman, I. *et al.* Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203* (2019).

7. Guo, B. *et al.* How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597* (2023).

8. He, X., Shen, X., Chen, Z., Backes, M. & Zhang, Y. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822* (2023).

9. Tian, Y. *et al.* Multiscale positive-unlabeled detection of ai-generated texts. *arXiv preprint arXiv:2305.18149* (2023).

10. Antoun, W., Mouilleron, V., Sagot, B. & Seddah, D. Towards a robust detection of language model generated text: Is chatgpt that easy to detect? *ArXiv* **abs/2306.05871** (2023).

11. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. 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)*, 4171–4186, DOI: 10.18653/v1/N19-1423 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019).

12. Liu, Y. *et al.* Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

13. Tian, E. Gptzero. https://gptzero.me/faq (2022).

14. Gehrmann, S., Strobelt, H. & Rush, A. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 111–116, DOI: 10.18653/v1/P19-3019 (Association for Computational Linguistics, Florence, Italy, 2019).

15. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D. & Finn, C. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305* (2023).

16. Su, J., Zhuo, T. Y., Wang, D. & Nakov, P. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540* (2023).

17. Abdelnabi, S. & Fritz, M. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. *2021 IEEE Symp. on Secur. Priv. (SP)* 121–140 (2020).

18. Grinbaum, A. & Adomaitis, L. The ethical need for watermarks in machine-generated language. *ArXiv* **abs/2209.03118** (2022).

19. Kirchenbauer, J. *et al.* A watermark for large language models. *ArXiv* **abs/2301.10226** (2023).

20. Krishna, K., Song, Y., Karpinska, M., Wieting, J. & Iyyer, M. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *ArXiv* **abs/2303.13408** (2023).

21. Conneau, A. *et al.* Unsupervised cross-lingual representation learning at scale. In *Annual Meeting of the Association for Computational Linguistics* (2019).

22. Bakhtin, A. *et al.* Real or fake? learning to discriminate machine from human generated text. *ArXiv* **abs/1906.03351** (2019).

23. Uchendu, A., Le, T., Shu, K. & Lee, D. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8384–8395, DOI: 10.18653/v1/2020.emnlp-main.673 (Association for Computational Linguistics, Online, 2020).

24. Raffel, C. *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21** (2020).

Acknowledgements

This study is supported by National Natural Science Foundation of China (No. 61772162), Key Projects of NSFC Joint Fund of China (No. U1866209), The Royal Society, International Exchanges 2018 Cost Share (China) Grant (No.IEC\NSFC\181300).

Author contributions statement

All authors conceived the experiments, H.X. conducted the experiments, All authors analyzed the results. All authors reviewed the manuscript.

Additional information

Competing interests: The authors declare no competing interests.