Received 14 February 2024, accepted 26 April 2024, date of publication 6 May 2024, date of current version 15 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3396820

RESEARCH ARTICLE

Enhancing Machine-Generated Text Detection: Adversarial Fine-Tuning of Pre-Trained Language Models

DONG HEE LEE, (Member, IEEE), AND BEAKCHEOL JANG, (Member, IEEE)
Graduate School of Information, Yonsei University, Seoul 03722, South Korea

Corresponding author: Beakcheol Jang (bjang@yonsei.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by Korean Government under Grant RS-2023-00273751.

ABSTRACT Advances in large language models (LLMs) have revolutionized the natural language processing field. However, the text generated by LLMs can result in various issues, such as fake news, misinformation, and social media spam. In addition, detecting machine-generated text is becoming increasingly difficult because it produces text that resembles human writing. We propose a new method for effectively detecting machine-generated text by applying adversarial training (AT) to pre-trained language models (PLMs), such as Bidirectional Encoder Representations from Transformers (BERT). We generated adversarial examples that appeared to have been modified by humans and applied them to the PLMs to improve the model's detection capabilities. The proposed method was validated on various datasets and experiments. It showed improved performance compared to traditional fine-tuning methods, with an average reduction in the probability of misclassification of machine-generated text by about 10%. We demonstrated the robustness of the model when generated with input tokens of different lengths and under different training data ratios. We suggested future research directions for applying AT to different languages and language model types. This study opens new possibilities for applying AT to the problem of machine-generated text detection and classification and contributes to building more effective detection models.

INDEX TERMS Machine generated text detection, text classification, adversarial training, large language models.

I. INTRODUCTION

Large language models (LLMs) have a significant impact on the field of natural language processing [1]. LLMs such as ChatGPT, Llama2, PaLM2, and GPT-4 [2], [3], [4], [5] can perform various tasks such as summarizing documents, translating, answering questions, and providing answers to complex questions across multiple domains. They are also used to improve the efficiency of daily work and life activities, such as content creation, programming code analysis, and writing.

Recent research indicates that text generated by LLMs can result in various issues, such as fake news, misinformation, and social media spam [6], [7]. Additionally, when students use LLMs to write assignments or essays, it can adversely affect their critical thinking and problem-solving abilities and reduce their motivation to learn [8]. LLMs can lead to issues such as plagiarized papers [9], manipulated public opinion [10], and malicious product reviews [11], all of which are recognized as important social issues.

Research on solving the problems caused by LLM-generated text and effectively distinguishing between machine and human-generated text is becoming increasingly important. The popularity of LLMs has led to a significant increase in the amount of text they generate, making it difficult for humans to differentiate between machine-generated and human-written text [12]. Therefore, there is a need for research on how to distinguish between the two automatically.

To address this issue, a method has been proposed for detecting machine-generated text by training the Robustly

The associate editor coordinating the review of this manuscript and approving it for publication was R. K. Tripathy.

optimized BERT pre-training Approach (RoBERTa) model [13], [14]. More recently, a zero-shot method was proposed for detecting machine-generated text by modifying it using the T5 [15] model and comparing it with the original [16]. However, supervised learning-based detection methods are limited by their vulnerability to attacks on text variations, and research is being conducted to address this issue [17].

Although generating adversarial examples is important, there is a scarcity of research focused on creating adversarial samples that resemble human modifications and applying them to detect machine-generated texts. The proposed method, inspired by DetectGPT [16], efficiently generates adversarial examples that resemble human modifications. We then apply adversarial training (AT) to language models such as Bidirectional Encoder Representations from Transformers (BERT) [18] to detect machine-generated text.

The main contributions of this study are summarized as follows: 1) We propose a novel method for generating adversarial examples that appear to be modified by humans and apply AT to language models such as BERT. Compared to fine-tuning the BERT model, the proposed method reduced the probability of model misclassification of machine-generated text by about 10% on average. It showed improved performance in terms of accuracy and F1 score. This demonstrates the superiority of the proposed methodology over existing methods. 2) Through experiments, we showed that the recognition performance decreases by approximately 2% as the length of the generated input sentences increases. This empirically shows that the more human text is written, the harder it is to detect machine-generated text.

The remainder of this study is organized as follows. Section II presents research on machine-generated text detection and AT. Section III explains how to fine-tune the pre-trained language model and the proposed methodology with AT. Section IV describes the datasets and hyperparameters used in the experiments and discusses the results obtained. Finally, the conclusions of this study and future research directions are presented in section V.

II. RELATED WORK

A. MACHINE GENERATED TEXT DETECTION

The emergence of various models [2], [3], [4], [5], which resulted from the development of LLMs, has significantly enhanced the ability to generate text across various domains. However, the texts generated were similar to those written by humans, making them difficult to distinguish [19].

The detection of machine-generated text is similar to text classification [20], [21]. One approach to address this issue is to use supervised learning, which involves a classification model trained to detect machine-generated text [22], [23], [24]. For instance, the Grover model proposed a method for detecting machine-generated text by adding a linear layer to a language model for fake news generation [25], and RoBERTa [13] proposed a classification model for detecting machine-generated text by training it [14]. However, these models can be vulnerable to attacks on textual variations, such as paraphrasing, which can degrade the detection performance. To address this issue, research has been conducted on classification models that use AT [17].

Statistics-based detection methods are crucial for identifying machine-generated texts. Giant Language Model Test Room (GLTR) [12] proposed a method to assist humans in detecting machine-generated text by analyzing word probability, rank, and entropy. In addition, a zero-shot-based detection method that uses the log probability of sentence tokens was proposed to identify machine-generated text [14]. The DetectGPT [16] model is an efficient zero-shot detection method for identifying machine-generated texts. This is achieved by perturbing the input text using a language model such as T5 [15] and comparing the original and perturbed text's log probabilities. Subsequent works, such as DetectLLM [21] and Fast-DetectGPT [26], improved upon the method proposed by DetectGPT.

B. ADVERSARIAL TRAINING

AT involves creating adversarial examples to induce errors in the model and training it using both these and original examples [27]. This approach enhanced the robustness of the model against adversarial attacks. AT has been studied in various fields, including image classification [28], [29], recommendation systems [30], and image generation [31], [32].

AT is extended to the text domain by applying it to word embeddings instead of perturbing the input vector of the Fast Gradient Sign Method (FGSM) [33]. In addition, AT has been applied to pre-trained language models that have shown excellent performance in various natural language processing tasks, such as natural language understanding, machine translation, and speech processing, by training large text data to understand the structure and context of language. In particular, because of the convergence of models such as BERT [18], XLNet [34], RoBERTa [13], T5 [15], and ELECTRA [35], research has been conducted to apply AT to the fine-tuning process of pre-trained language models. Specifically, AT methods that perturb the text embedding layer have been employed to enhance text classification performance [36]. AT has also been applied to the pre-training of language models [37], and fine-tuning methods incorporating both AT and regularization have been used to improve the generalization performance of pre-trained language models [38], [39]. Moreover, empirical evidence demonstrates the effectiveness of AT when applied to a BERT model [40].

AT has been utilized in detecting machine-generated text, including the RADAR [17] model method for detecting machine-generated text that is robust against attacks such as paraphrasing. Furthermore, AT has been proposed for detecting fake news [41]. This study differs from previous ones in that it generates adversarial examples similar to those written by humans.

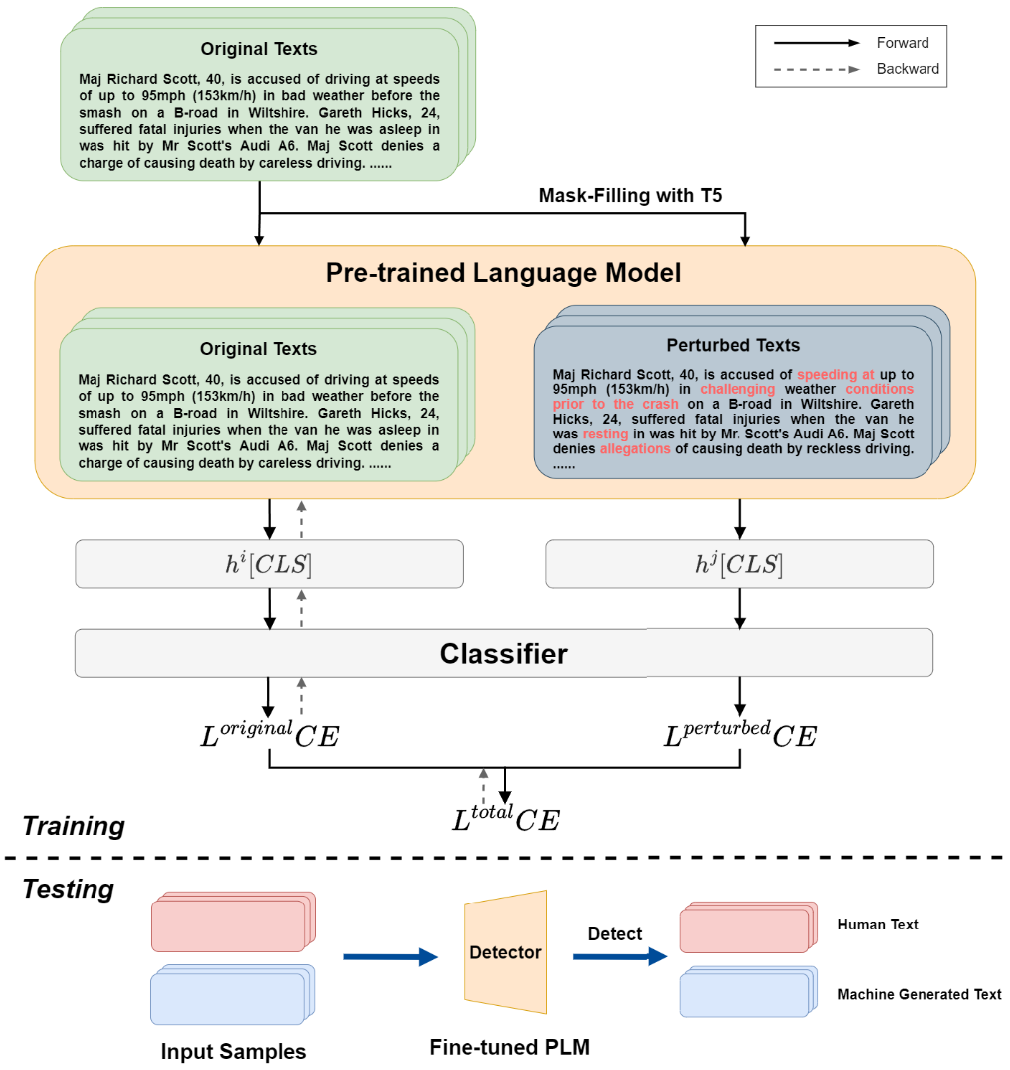
Original Texts
Maj Richard Scott, 40, is accused of driving at speeds of up to 95mph (153km/h) in bad weather before the smash on a B-road in Wiltshire. Gareth Hicks, 24, suffered fatal injuries when the van he was asleep in was hit by Mr Scott's Audi A6. Maj Scott denies a charge of causing death by careless driving. ......
Forward
Backward
Mask-Filling with T5
Pre-trained Language Model
Perturbed Texts
Maj Richard Scott, 40, is accused of speeding at up to 95mph (153km/h) in challenging weather conditions prior to the crash on a B-road in Wiltshire. Gareth Hicks, 24, suffered fatal injuries when the van he was resting in was hit by Mr. Scott's Audi A6. Maj Scott denies allegations of causing death by reckless driving. ......
$h^i[CLS]$
$h^j[CLS]$
Classifier
$L^{original}CE$
$L^{perturbed}CE$
$L^{total}CE$
Training
Testing
Detector
Detect
Human Text
Machine Generated Text
Input Samples
Fine-tuned PLM

FIGURE 1. Overview of the proposed method. Our proposed model architecture aims to fine-tune pre-trained language models for machine-generated text classification tasks. We use the T5 model to perturb the input text by masking and filling to generate adversarial examples. The model is then trained using Cross Entropy Loss on both the original and adversarial examples.

III. PROPOSED METHOD

This study aims to utilize AT on pre-trained language models, such as BERT, to develop detection models that can effectively identify machine-generated text. In this section, we describe the process of fine-tuning a pre-trained language model such as BERT for a text classification task. In addition, the proposed method for applying AT to language models by generating adversarial examples, Figure 1 shows an overview of the proposed method.

A. PRE-TRAINED LANGUAGE MODEL FINE-TUNING

We aim to binarily classify inputs, such as $\{x_i, y_i\}_{i=1,\ldots,N}$ using a transformer-based pre-trained language model (PLM), such as BERT. A pre-trained language model such as (1) is given an input token sequence x_i = [CLS, $t_1, \ldots, t_T$, SEP]. It outputs representation information, such as $H^L = h^L_{[\mathrm{CLS}]}, h^L_1, \ldots, h^L_T, h^L_{[\mathrm{SEP}]}$, where L is the number of model layers.

$$h^L_{[\mathrm{CLS}]}, h^L_1, \ldots,, h^L_T, h^L_{[\mathrm{SEP}]} = \mathrm{PLM}([\mathrm{CLS}], t_1, \ldots, t_T, [\mathrm{SEP}]) \quad (1)$$

The most common method to fine-tune such a PLM is to add a softmax classifier to the sentence-level representations of the model, such as $h_{[\mathrm{CLS}]}$, which is the final hidden state of BERT's [CLS] token, as shown in (2): Let $W \in \mathbb{R}^{dc \times d_h}$, where C is the number of classes.

$$p(y_c|h_{[\mathrm{CLS}]}) = \mathrm{softmax}(Wh_{[\mathrm{CLS}]}) \quad c \in C \quad (2)$$

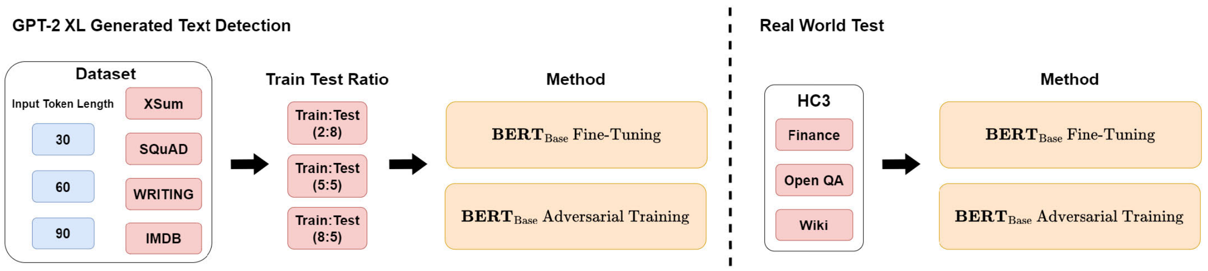

FIGURE 2. The overall experimental design. To compare the performance of machine-generated text detection using the BERT model, performance was evaluated with different training and test data ratios (2:8, 5:5, 8:2) using text generated with different input token lengths (30, 60, 90). Real-world applicability was evaluated using the HC3 dataset containing ChatGPT answers and human answers.

The model was trained to minimize the cross-entropy loss, as shown in (3), where N is the batch size.

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(p(y_{i,c}|h^{i}_{[\mathrm{CLS}]})) \tag{3}$$

B. ADVERSARIAL TRAINING FOR MACHINE GENERATED TEXT DETECTOR

AT is a model training method using both original examples x_i and adversarial examples $x_i + r$ by creating adversarial examples that introduce small imperceptible perturbations into the input, which cause misclassification. The advantage of this method is that it enables us to build a robust model. Therefore, we propose applying an AT approach using both original and adversarial examples to fine-tune a pre-trained language model to classify machine-generated text.

Algorithm 1 Adversarial Training for Binary Text Classification With Pre-Trained Language Models

```
1:  Input: Dataset $D = \{(x_i, y_i)\}$ for $i = 1$ to $N$
2:  Pre-trained BERT parameters $\theta$
3:  Pre-trained T5-small parameters $\phi$ for adversarial example generation
4:  Learning rate $\eta$, number of epochs $T$, adversarial loss weight $\alpha$
5:  Output: Fine-tuned BERT parameters $\theta^*$
6:  for $epoch = 1$ to $T$ do
7:      for $(x, y)$ in $D$ do
8:          Generate adversarial example $x_{per}$ using $\phi$ with mask-filling
9:          $p \leftarrow \sigma(\text{BERT Classifier}(x; \theta))$
10:         $p_{per} \leftarrow \sigma(\text{BERT Classifier}(x_{per}; \theta))$
11:         $L \leftarrow -[y\log(p) + (1-y)\log(1-p)]$
12:         $L_{per} \leftarrow -[y\log(p_{per}) + (1-y)\log(1-p_{per})]$
13:         $L_{total} \leftarrow L + \alpha L_{per}$
14:         $\theta \leftarrow \theta - \eta\nabla_{\theta}L_{total}$
15:     end for
16: end for
17: $\theta^* \leftarrow \theta$
```

The classification of machine-generated text can be viewed as a binary text classification problem. Generated sentences are typically modified from the original content, which can be considered perturbations to the input. We propose a new approach for fine-tuning pre-trained language models, such as BERT, to classify machine-generated text. This approach enhances detection performance by leveraging the nature of AT, enabling the model to identify both original and modified text. In addition, this approach uses mask-filling language models such as T5 to generate adversarial examples for AT.

We applied loss functions to the original and perturbed data during training. The total loss function is defined as the weighted sum of the original loss $L_{original}$ and the perturbed loss $L_{perturbed}$, expressed as $L_{total} = L_{original} + \alpha * L_{perturbed}$. In this study, we set the value of α to 0.1.

IV. EXPERIMENT

A. EXPERIMENT DESIGN

We proposed to apply adversarial learning in building classification models to detect machine-generated text from pre-trained language models. Adversarial examples were generated like human corrections and were used for adversarial learning. This study conducted experiments under different conditions to compare the detection performance of proposed and existing methods for fine-tuning pre-trained language models. The experimental design is illustrated in Fig. 2.

The experiment compared the detection performances using text generated by the GPT-2 XL model. The text generated by GPT-2 XL was generated with different input token lengths (30, 60, and 90) and used four datasets: XSum, SQuAD, WRITING, and IMDB. Referring to the input token length of 30 employed by DetectGPT [16] for text generation, we wanted to determine how the detection performance changed for texts generated under different conditions. The pre-trained language model used the BERT model, and we compared the detection performance of machine-generated text using fine-tuning and adversarial learning. During the training process, we set the training and test data ratios to 2:8, 5:5, and 8:2 to observe the changes in the performance with different training data ratios.

In addition, to compare the detection performance of the proposed adversarial learning method and the traditional fine-tuning method in the real world and to evaluate their applicability, we used the HC3 dataset [42]. The HC3 dataset comprises ChatGPTs and human responses to questions in

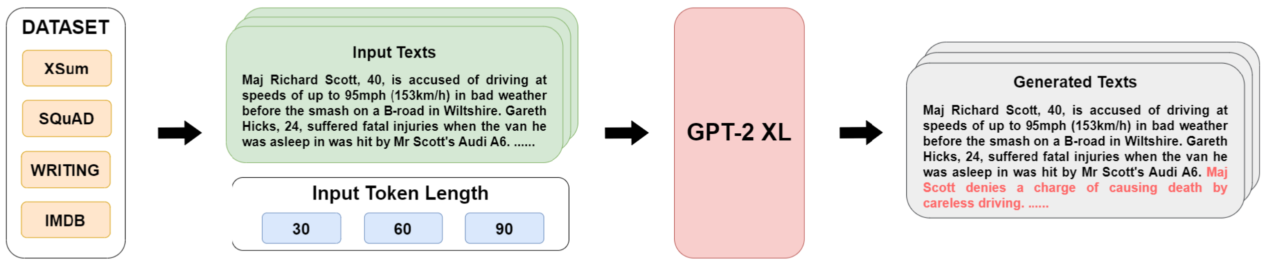
DATASET
XSum
SQuAD
WRITING
IMDB
Input Texts
Maj Richard Scott, 40, is accused of driving at speeds of up to 95mph (153km/h) in bad weather before the smash on a B-road in Wiltshire. Gareth Hicks, 24, suffered fatal injuries when the van he was asleep in was hit by Mr Scott's Audi A6. ......
Input Token Length
30
60
90
GPT-2 XL
Generated Texts
Maj Richard Scott, 40, is accused of driving at speeds of up to 95mph (153km/h) in bad weather before the smash on a B-road in Wiltshire. Gareth Hicks, 24, suffered fatal injuries when the van he was asleep in was hit by Mr Scott's Audi A6. Maj Scott denies a charge of causing death by careless driving. ......

FIGURE 3. Experimental Process of Text Generation. This experiment used the GPT-2 XL model to generate text from four datasets: XSum, SQuAD, WRITING, and IMDB. The process was conducted by inputting tokens of 30, 60, and 90 lengths from the beginning of texts in each dataset.

TABLE 1. Experimental results for detecting text generated by GPT-2 XL. Results were summarized using the average value for each experimental condition. The proposed method outperforms the existing fine-tuning methods on most datasets. In the table, the best performance scores are highlighted in bold.

Metric	Method	XSum	SQuAD	WRITING	IMDB	AVG
Acc	$\text{BERT}_{\text{Base}}$ Fine-Tuning	0.9563	**0.9561**	0.9656	0.9434	0.9553
	$\text{BERT}_{\text{Base}}$ AT(ours)	**0.9625**	0.9540	**0.9776**	**0.9462**	**0.9601**
F1	$\text{BERT}_{\text{Base}}$ Fine-Tuning	0.9552	**0.9555**	0.9654	0.9433	0.9548
	$\text{BERT}_{\text{Base}}$ AT(ours)	**0.9616**	0.9528	**0.9775**	**0.9453**	**0.9593**

various domains. For the real-world test, we used 80% of the total data as training data and evaluated the recognition performance using 20% of the data without adjusting the training ratio.

B. DATASET

This study evaluates the performance of classification models for classifying machine-generated text using various datasets. This study used news articles from XSum [43], context from SQuAD [44], prompts from WRITING [45], and reviews from IMDB [46]. 5,000 sentences were extracted from each dataset. The GPT-2 XL model generated machine-generated text using only a certain number of tokens from the beginning of the original sentence as input. To generate sentences, we set the number of input tokens to 30, 60, and 90 and limited the maximum sentence length to 512.

An additional 5,000 texts were selected from each dataset, excluding those used for machine-generated text. This resulted in two datasets containing equal positive and negative examples, totalling 10,000 texts each. The text generation design is illustrated in Fig. 3.

C. EVALUATION METRIC

We evaluated the model's performance using the Accuracy and F1-score, commonly used to evaluate the performance of classification problems. Accuracy is defined as the proportion of actual and predicted values that match, as in (4), while the F1-score is the harmonic mean of the model's precision and recall, as in (5). True positives (TP) are values correctly identified as positive, and true negatives (TN) are correctly identified as negative. False positives (FP) are values incorrectly identified as positive, and false negatives (FN) are values incorrectly identified as negative. The F1-Score and accuracy were calculated defined as follows.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$\text{F1} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{5}$$

D. TRAINING DETAILS

Experiments were conducted to compare the performance of the proposed method with that of existing methods for fine-tuning BERT-based models using machine-generated sentences. The experiments were performed using an NVIDIA GeForce RTX 3090 GPU. All models were fine-tuned for two epochs using a random seed of zero, with a batch size of eight and a learning rate of 1e-4. An Adam Optimizer was used, and a linear scheduler was applied for efficient learning. The T5-small model was also used to produce adversarial examples. Additionally, sentences were regenerated by masking words with 2-grams for 30% of the total sentences.

E. EXPERIMENT RESULT

We propose a method for fine-tuning pre-trained language models using AT. We generate adversarial examples that appear to be modified by humans using the T5 model to classify machine-generated text. Experiments were conducted to compare the performance of the proposed method with that of existing methods by varying the length of the input tokens used for text generation and the training-data ratio. The average detection performance for each input token length and training data ratio is summarized in Table 1.

The experimental results indicate that the proposed method enhances the accuracy and F1-score on all datasets except SQuAD. Compared to the fine-tuning method, the

TABLE 2. Experimental results as a function of input token length used for generation. The input token lengths of 30, 60, and 90 were used, and the average of the Train Test Ratio experimental results were analyzed. In the table, the best performance scores are highlighted in bold.

Metric	Method	XSum			SQuAD			WRITING			IMDB		
		30	60	90	30	60	90	30	60	90	30	60	90
ACC	$\text{BERT}_{\text{Base}}$ Fine-Tuning	0.9749	0.9617	0.9323	0.9684	**0.9524**	**0.9475**	0.9781	0.9749	0.9438	0.9614	0.9453	**0.9235**
	$\text{BERT}_{\text{Base}}$ AT(ours)	**0.9770**	**0.9675**	**0.9430**	**0.9698**	0.9484	0.9437	**0.9847**	**0.9829**	**0.9653**	**0.9671**	**0.9508**	0.9208
F1	$\text{BERT}_{\text{Base}}$ Fine-Tuning	0.9745	0.9609	0.9300	0.9679	**0.9517**	**0.9468**	0.9784	0.9752	0.9425	0.9614	0.9461	**0.9224**
	$\text{BERT}_{\text{Base}}$ AT(ours)	**0.9768**	**0.9668**	**0.9412**	**0.9691**	0.9473	0.9418	**0.9848**	**0.9828**	**0.9650**	**0.9670**	**0.9498**	0.9190

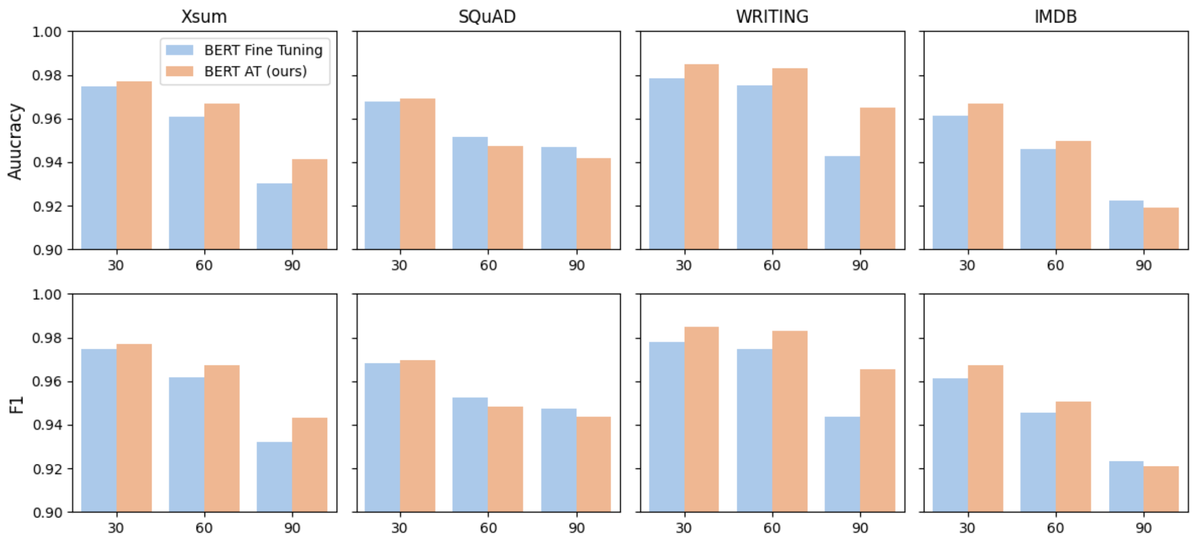

FIGURE 4. Detection performance of machine-generated output varies based on input token length. The graph displays the performance of machine-generated detection based on the length of input tokens. The x-axis shows the input token length, with 30, 60, and 90 values. The y-axis represents accuracy and F1-score. The results indicate that as the length of input tokens increases, both accuracy and F1 score decrease. This trend is consistent across all datasets.

proposed method improved the probability of misclassifying machine-generated text by about 10%, with an average increase in accuracy and F1-score of about 1% and a maximum performance improvement of 1.3%. These results demonstrate that our method with AT can achieve superior performance for machine-generated text classification compared to traditional fine-tuning methods.

F. GENERATE INPUT TOKEN LENGTH RESULT

We experimented to analyze the effect of the number of input tokens used in text generation on the machine-generated text detection performance. We generated machine-generated text under different conditions by setting the number of input tokens to 30, 60, and 90 when the text was generated using the GPT-2 XL model. The experimental results are summarized in Table 2. Result shows that the proposed method outperforms existing fine-tuning methods regarding accuracy and F1-score in most cases.

As shown in Figure 4, we observed a gradual decrease in the detection performance as the number of input tokens used in the generation process increased. On average, the detection performance decreased by approximately 2% as the length of the input token used for generation increased, which means that as the length of the input token increased, the proportion of text written or modified by humans increased, making it more challenging to detect machine-generated text accurately. The method proposed in this study exhibited high robustness against performance degradation under different generation conditions. Specifically, when using the traditional fine-tuning method on the WRITING dataset, the accuracy decreased by 3.5% from 0.9781 to 0.9438, and the F1-score decreased by 3.7% from 0.9784 to 0.9425. However, for the proposed method, the accuracy decreased by 2% from 0.9847 to 0.9653, and the F1-score decreased by 2% from 0.9848 to 0.9650. These results show that our adversarial learning method is more robust and effective than the existing methods for detecting generated text under different conditions.

G. TRAIN TEST RATIO RESULT

We conducted experiments using different training and test data ratios to analyze how the amount of training data affects the performance of machine-generated text detection. Specifically, we set the ratio of the training to test data after fine-tuning to 2:8, 5:5, and 8:2, respectively. The experimental results are listed in Table 3.

This study found that the proposed method outperformed existing fine-tuning methods by about 1% on average under

TABLE 3. Experimental results as a function of the ratio of training data used for training. We split the training and test data in ratios 2:8, 5:5, and 8:2 and used the average of the detection performance as a function of input token length. In the table, the best performance scores are highlighted in bold.

Metric	Method	XSum			SQuAD			WRITING			IMDB		
		2:8	5:5	8:2	2:8	5:5	8:2	2:8	5:5	8:2	2:8	5:5	8:2
ACC	BERT$_{\text{Base}}$ Fine-Tuning	0.9403	0.9595	0.9690	**0.9546**	**0.9512**	0.9625	0.9587	0.9675	0.9707	**0.9308**	0.9459	0.9533
	BERT$_{\text{Base}}$ AT(ours)	**0.9453**	**0.9667**	**0.9755**	0.9432	0.9511	**0.9677**	**0.9655**	**0.9803**	**0.9870**	0.9283	**0.9489**	**0.9615**
F1	BERT$_{\text{Base}}$ Fine-Tuning	0.9383	0.9589	0.9683	**0.9545**	**0.9503**	0.9617	0.9583	0.9678	0.9700	**0.9320**	0.9453	0.9527
	BERT$_{\text{Base}}$ AT(ours)	**0.9439**	**0.9658**	**0.9751**	0.9417	0.9495	**0.9671**	**0.9651**	**0.9804**	**0.9871**	0.9261	**0.9493**	**0.9605**

TABLE 4. HC3 experimental results. The proposed method outperformed the fine-tuning method on the Finance and Wiki datasets, except for the Open QA data. In the table, the best performance scores are highlighted in bold.

Metric	Method	Finance	Open QA	Wiki
Acc	BERT$_{\text{Base}}$ Fine-Tuning	0.9797	**0.9979**	0.9644
	BERT$_{\text{Base}}$ AT(ours)	**0.9949**	0.9937	**0.9852**
F1	BERT$_{\text{Base}}$ Fine-Tuning	0.9794	**0.9979**	0.9653
	BERT$_{\text{Base}}$ AT(ours)	**0.9949**	0.9937	**0.9853**

different training data ratios. These results indicate that the proposed method can effectively learn from relatively limited data and be useful, even when using a small dataset.

The experimental results also showed that detection performance improved as the data used for training increased. For example, when comparing the performance when 80% of the total data was used for training versus only 20%, we found that the detection performance decreased by an average of 2.4%.

H. REAL WORLD TEST

In the previous experimental phase, machine-generated text detection was performed using sentences generated by GPT-2 XL. To evaluate the applicability of the proposed method in real-world settings, we conducted experiments using the Human ChatGPT Comparison Corpus (HC3) dataset was used to conduct experiments.

The HC3 dataset comprises data from English and Chinese. The English dataset contained human and ChatGPT responses to approximately 24,000 questions from the various datasets, including Finance, Open QA, Wiki, Medicine, and Reddit. In this study, we compared the detection performance of the proposed method with existing methods using data from finance, Open QA, and Wiki.

The experiments used 7,866 Finance, 1,684 Wiki, and 2,374 Open QA sentences. Each dataset's 80% was used for training, and 20% was used for evaluation. The experimental results are summarized in Table 4. The proposed method outperformed existing methods by an average of 1% on the Finance and Wiki datasets, except for the Open QA dataset. The accuracy and F1-score improved by 1.6% for each Finance dataset. On the Wiki dataset, the proposed method improved the accuracy and F1-score by 2.2% and 2.1%, respectively, compared to the existing methods. These results demonstrate the effectiveness of the proposed method for detecting machine-generated sentences in real-world settings.

V. CONCLUSION

This study presents a novel approach for detecting text produced by large language models and validates its efficacy through experiments. The proposed method applies AT to language models by generating examples that resemble human modifications. The proposed method demonstrated an average performance improvement of approximately 10% compared to existing fine-tuning methods. It also shows robustness under different input token lengths and training-data ratios. These results indicate the potential of the new approach in machine-generated text detection. They are expected to be widely applied in future research and applications related to machine-generated text detection.

However, this study had some limitations. First, since the proposed method uses AT methods focused on the BERT model, it is necessary to conduct comparative experiments using various language models. Second, exploring different techniques and methodologies to generate adversarial examples is necessary because we only utilized the T5 model. Third, it is necessary to conduct an in-depth analysis of the proposed method using a variety of evaluation metrics and ablation studies. Finally, this study was limited to linguistic diversity, using only an English dataset for experiments.

Therefore, future research should apply AT methods to different types of language models and propose a generalized approach with validation. In addition, multilingual datasets and a variety of evaluation metrics should be employed to validate the effectiveness of the AT approach. Finally, a crucial area for future research is the application of various recent models and techniques to generate adversarial examples and propose ways to respond to more sophisticated and diverse types of adversarial attacks. These studies are expected to significantly contribute to expanding the scope of research on LLM-generated text detection.

REFERENCES

[1] W. Xin Zhao et al., "A survey of large language models," 2023, *arXiv:2303.18223*.

[2] OpenAI. (2022). *ChatGPT: Optimizing Language Models for Dialogue*. [Online]. Available: https://openai.com/blog/chatgpt

[3] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.

[4] R. Anil et al., "PaLM 2 technical report," 2023, *arXiv:2305.10403*.

[5] OpenAI. (2023). *GPT-4 Technical Report*. [Online]. Available: https://ar5iv.org/abs/2303.08774

[6] A. Al Ayub Ahmed, A. Aljabouh, P. Kumar Donepudi, and M. Suh Choi, "Detecting fake news using machine learning : A systematic literature review," 2021, *arXiv:2102.04458*.

[7] E. N. Crothers, N. Japkowicz, and H. L. Viktor, “Machine-generated text: A comprehensive survey of threat models and detection methods,” *IEEE Access*, vol. 11, pp. 70977–71002, 2023.
[8] E. Kasneci et al., “ChatGPT for good? On opportunities and challenges of large language models for education,” *Learn. Individual Differences*, vol. 103, Apr. 2023, Art. no. 102274.
[9] J. Philip Wahle, T. Ruas, F. Kirstein, and B. Gipp, “How large language models are transforming machine-paraphrased plagiarism,” 2022, *arXiv:2210.03568*.
[10] H. Stiff and F. Johansson, “Detecting computer-generated disinformation,” *Int. J. Data Sci. Analytics*, vol. 13, no. 4, pp. 363–383, May 2022.
[11] D. I. Adelani, H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi, and I. Echizen, “Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection,” in *Proc. 34th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*. Cham, Switzerland: Springer, 2020, pp. 1341–1354.
[12] S. Gehrmann, H. Strobelt, and A. M. Rush, “GLTR: Statistical detection and visualization of generated text,” 2019, *arXiv:1906.04043*.
[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” 2019, *arXiv:1907.11692*.
[14] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. Wook Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, and J. Wang, “Release strategies and the social impacts of language models,” 2019, *arXiv:1908.09203*.
[15] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.
[16] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, “DetectGPT: Zero-shot machine-generated text detection using probability curvature,” 2023, *arXiv:2301.11305*.
[17] X. Hu, P.-Y. Chen, and T.-Y. Ho, “RADAR: Robust AI-text detection via adversarial learning,” 2023, *arXiv:2307.03838*.
[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv:1810.04805*.
[19] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, “Automatic detection of generated text is easiest when humans are fooled,” 2019, *arXiv:1911.00650*.
[20] G. Jawahar, M. Abdul-Mageed, and L. V. S. Lakshmanan, “Automatic detection of machine generated text: A critical survey,” 2020, *arXiv:2011.01314*.
[21] J. Su, T. Yue Zhuo, D. Wang, and P. Nakov, “DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text,” 2023, *arXiv:2306.05540*.
[22] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, and A. Szlam, “Real or fake? Learning to discriminate machine from human generated text,” 2019, *arXiv:1906.03351*.
[23] A. Uchendu, T. Le, K. Shu, and D. Lee, “Authorship attribution for neural text generation,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 8384–8395.
[24] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi, “TweepFake: About detecting deepfake tweets,” *PLoS ONE*, vol. 16, no. 5, May 2021, Art. no. e0251415.
[25] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, “Defending against neural fake news,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–2.
[26] G. Bao, Y. Zhao, Z. Teng, L. Yang, and Y. Zhang, “Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature,” 2023, *arXiv:2310.05130*.
[27] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, *arXiv:1412.6572*.
[28] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.
[29] J. Su, D. V. Vargas, and K. Sakurai, “One pixel attack for fooling deep neural networks,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
[30] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, “Data poisoning attacks on factorization-based collaborative filtering,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.
[31] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[32] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.
[33] T. Miyato, A. M. Dai, and I. Goodfellow, “Adversarial training methods for semi-supervised text classification,” 2016, *arXiv:1605.07725*.
[34] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized autoregressive pretraining for language understanding,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.
[35] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training text encoders as discriminators rather than generators,” 2020, *arXiv:2003.10555*.
[36] L. Pan, C.-W. Hang, A. Sil, and S. Potdar, “Improved text classification via contrastive adversarial training,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 10, pp. 11130–11138.
[37] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, and J. Gao, “Adversarial training for large neural language models,” 2020, *arXiv:2004.08994*.
[38] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao, “SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization,” 2019, *arXiv:1911.03437*.
[39] H. Cheng, X. Liu, L. Pereira, Y. Yu, and J. Gao, “Posterior differential regularization with f-divergence for improving model robustness,” 2020, *arXiv:2010.12638*.
[40] J. Ebrahimi, H. Yang, and W. Zhang, “How does adversarial fine-tuning benefit BERT?” 2021, *arXiv:2108.13602*.
[41] A. Tariq, A. Mehmood, M. Elhadef, and M. U. G. Khan, “Adversarial training for fake news classification,” *IEEE Access*, vol. 10, pp. 82706–82715, 2022.
[42] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, and Y. Wu, “How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection,” 2023, *arXiv:2301.07597*.
[43] S. Narayan, S. B. Cohen, and M. Lapata, “Don’t give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization,” 2018, *arXiv:1808.08745*.
[44] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” 2016, *arXiv:1606.05250*.
[45] A. Fan, M. Lewis, and Y. Dauphin, “Hierarchical neural story generation,” 2018, *arXiv:1805.04833*.
[46] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Hum. Lang. Technol.*, Portland, OR, USA, Jun. 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

DONG HEE LEE (Member, IEEE) received the B.S. degree in business administration and industrial engineering from Dongguk University, Seoul, South Korea, in 2023. He is currently pursuing the M.S. degree in information systems with Yonsei University, Seoul, South Korea. His research interests include deep learning, NLP, and causal machine learning.

BEAKCHEOL JANG (Member, IEEE) received the B.S. degree in computer science from Yonsei University, in 2001, the M.S. degree in computer science from Korea Advanced Institute of Science and Technology, in 2002, and the Ph.D. degree in computer science from North Carolina State University, in 2009. He is currently a Professor with the Graduate School of Information, Yonsei University. His primary research interests include artificial intelligence, bigdata analytics, and natural language processing. He is a member of ACM.