Su Zehao

58818678

Comparative Analysis of Pre-trained Transformer Models for Early Detection of AI-Generated Text in Online Media

supervisor name: Wong Ka Chun

Abstract

In recent years, the popularity of large-scale language modeling has led to an increase in the content of AI-generated texts, and thus there is an urgent need for robust and reliable detection methods. In this paper, we propose a new detection framework that integrates the Dual-Text Difference Feature (DTDF) and Dual-Stream Feature Fusion (DSFF) architectures to efficiently differentiate between AI- and human-written texts. Specifically, DTDF captures text revisions at the semantic level by calculating the absolute difference between sentence embeddings of the original and AI-revised versions using the Paraphrase-MiniLM-L6-v2 model. This difference representation highlights subtle semantic changes that often occur in AI-generated revisions. The DSFF architecture jointly models the original semantic embeddings and DTDF through parallel LSTM branching, and then further improves detection performance through a unified classification layer.

We conducted comprehensive experiments across multiple pre-trained language models based on OpenGPTText dataset, including BERT, RoBERTa, GPT-2, and MiniLM. The results consistently demonstrate that, except in the case of GPT-2, differential features outperform raw semantic features in classification accuracy, precision, recall, F1-score, and AUC. Furthermore, models built on the DSFF framework demonstrate the highest performance across all metrics, confirming the effectiveness of our proposed architecture. Notably, RoBERTa shows superior results when modeling raw semantics, while paraphrase-MiniLM-L6-v2 excels in capturing differential semantics and performs best overall under the DSFF configuration.

These findings validate the usefulness of semantic difference modeling for detecting AI-generated text and highlight the practical value of dual-stream architectures for capturing the nuanced linguistic patterns introduced by generative models.

Keywords: AI-generated text detection; Dual-Text Differential Features; Dual-Stream Feature Fusion; Paraphrase-MiniLM-L6-v2 model; OpenGPTText dataset

1. Introduction

Recently, Natural Language Processing (NLP) has really changed a lot, thanks to the quick progress in Large Language Models (LLMs) like BERT, T5, and GPT. These large-scale unsupervised pre-trained models have demonstrated unprecedented capabilities in language understanding and generation, greatly facilitating research and applications in areas such as machine translation, automatic summarization, question answering and dialogue systems [1 - 3]. Among them, ChatGPT, developed by OpenAI based on the GPT-3 and GPT-4 architectures, has attracted much attention for its versatile capabilities in multi-round dialogue, logical reasoning and code generation. It has rapidly accumulated tens of millions of active users, marking a critical step towards the practical deployment of Artificial General Intelligence (AGI).

However, this technological advancement also raises serious ethical and security concerns and can be considered a modern metaphor for Pandora's Box. Current research suggests that malicious actors can take advantage of the openness and ease of use of LLMs to automatically generate large amounts of disinformation and propaganda [4], fabricate academic papers and application materials, manipulate public opinion, and operate botnets. In addition, LLMs can be used to mimic the linguistic style of specific individuals for cyber fraud and identity fraud. The content generated by these models may also infringe intellectual property rights by making unauthorized copies of copyrighted material. On e-commerce platforms, AI-generated reviews may mislead consumers and undermine fair competition [5]. A recent study by Carlini et al [6] pointed out a bigger problem. They found that using
LLMs to create fake training data can lead to a cycle where this data is reused to train new models. This creates a kind of feedback loop, or a "data contamination loop", which could cause bigger issues in the future. This cycle threatens the ability of models to generalize and poses a structural challenge to the trustworthiness of machine learning systems.

Thus, the core challenge for NLP and other AI fields is twofold: first, how to fully utilize the potential of LLM while reducing its misuse; and second, how to ensure the integrity of digital content and build public trust. Developing innovative text-based AI testing techniques has become a top priority for researchers and practitioners [7]. This work requires not only the development of highly accurate algorithmic solutions, but also the consideration of privacy, cross-linguistic adaptability, stability and fairness. Recognizing AI-generated text remains a challenging task. As the quality of text generated by Large Language Models (LLMs) continues to improve, the differences between AI-generated content and human-authored content in terms of linguistic style, logical structure, and semantic fluency are rapidly shrinking, rendering traditional methods based on human judgement virtually ineffective. Studies have shown that, in the absence of external tools, the accuracy of human recognition of AI-generated text is only slightly better than random guessing.

Therefore, there is an urgent need to develop novel algorithms capable of accurately distinguishing between machine and human generated text. Our research aims to create a robust classifier capable of distinguishing between human-generated text and AI-generated text. The classifier will use an innovative dual-stream framework to extract discriminative features from both the original text and the AI revised text. We plan to analyze these features and train the model to correctly identify text sources. We evaluated various complex models on the

OpenGPText-English dataset, including CNN-LSTM, BERT, T5, GPT-2, RoBERTa, and Paraphrase MiniLM L6-v2. We found that the two types of features extracted by the dual-stream framework yielded good results in both models. Among them, the Paraphrase-MiniLM-L6-v2 model produces the best results. The key contributions of this paper are outlined as follows:

- Feature Engineering lies in the novel introduction of differential features derived from AI self-revision behavior. This work proposes, a Dual-Text Differential Feature Extraction framework that leverages the behavioral signals of AI-generated text revisions. Specifically, an AI model is prompted (e.g., “revise the following text”) to generate a revised version of the input, and a pretrained semantic encoder (paraphrase-MiniLM-L6-v2) is employed to quantify the semantic shift between the original and revised texts. A set of Revision-Sensitive Features is then designed, encompassing vocabulary-level dynamics (e.g., vocabulary change ratio), syntactic alterations (e.g., sentence count ratio), surface-level token changes (e.g., punctuation/capitalization variations), and multi-granularity similarity measures (e.g., word/character-level similarity), providing behavior-informed and discriminative representations for downstream detection tasks.
- Model Architecture is embodied in the design of a Dual-Stream Feature Fusion framework, which departs from conventional single-stream paradigms by enabling synergistic perception of textual and behavioral features. The model consists of two parallel processing streams: an original feature stream that captures intrinsic semantic representations via pretrained encoders (e.g., BERT or RoBERTa), and a differential feature stream that focuses on behavioral signals extracted from the contrast between the original and revised texts. These streams are fused through a dynamic feature fusion layer via concatenation, allowing the classifier to jointly optimize over both static textual semantics and dynamic revision-induced cues. This dual-stream design significantly enhances the model’s ability to detect and distinguish AI-generated content with greater robustness and interpretability.

The rest of this paper is structured as follows: We first discuss related work in Section 2; illustrate OpenGPTText data set and present our model and the training details in Section 3; evaluate the performance using various metrics and designs ablation experiments based on the Dual-Stream Feature Fusion architecture and analyzes the experimental results in Section 4. Finally conclude in Section 5.

2. Related Work

This chapter will review existing research on AIgenerated text and explore the classification of texts generated by AI and humans. The two main types of AI-based model evaluation techniques currently in use are Prepared Detectors and Post-hoc Detectors. While the latter depends on a retrospective analysis the content of model generated, the former stresses the proactive integration of evaluation mechanisms during the model generation phase, as illustrated in Figure 1. Every strategy has unique advantages and disadvantages.

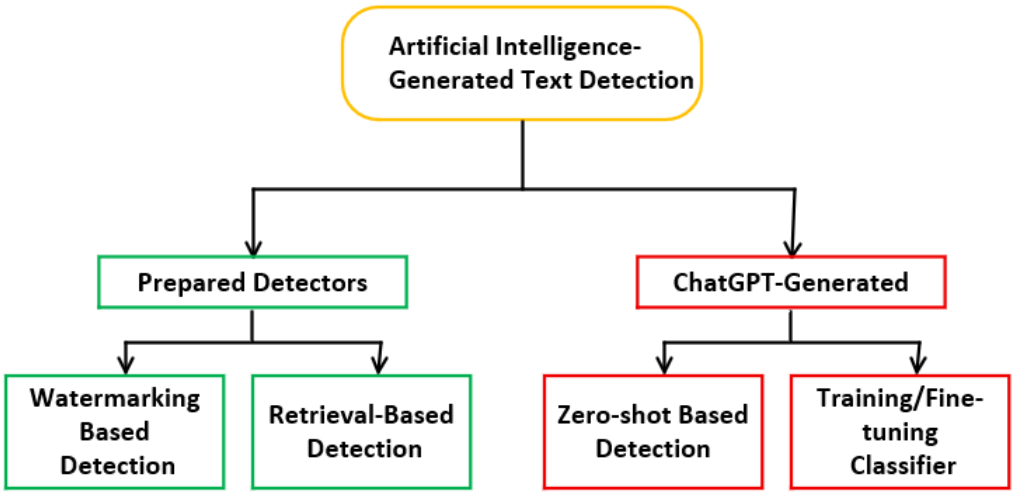


Figure1: Categorization of all explained methods

Prepared detectors are a technique for inserting identifiable information into a language model during its text generation. These methods generally require model developers to intervene during the text generating process by inserting information or features that are then extracted during the testing process. Considering the configuration of the embedded information and the corresponding testing mechanisms, the main features of the prepared detector are as follows:

- Watermarking methods [8 - 10]: These approaches introduce imperceptible markers into generated text by manipulating the model’ s token sampling distribution—such as biasing toward certain vocabulary — or through guided word substitutions. For instance, GPTWatermark utilizes a pseudorandom method to select labels "green list," increasing the probability of their inclusion, thereby embedding watermark signals that can be systematically identified in the output.
- Retrieval-based methods [11]: These techniques store both the generated content and its corresponding prompts during the generation process and then detect the AI-generated text by calculating the semantic similarity between the two. This approach is particularly effective in controlled environments where the platform has full access to prompt-response records.

The advantages of these methods are high accuracy, robustness and adaptability to the generation model. The effectiveness of these methods is especially important in settings

where the developer has complete control over the environment. However, an important limitation of these methods is the heavy reliance on model collaboration. When it comes to open-source models or models generated through third-party platforms, these methods are often unavailable, thus greatly limiting their general applicability.

The post-hoc detector, on the other hand, is independent of any embedding information. Instead, it decides whether the generated text is from an AI model based on its linguistic features and structure. Since these techniques don't necessitate access to or change of the generated model, they are ideal for a wide range of real-world uses, including social media content management, academic paper review, and news verification. These methods have generated a lot of interest and have been used in a variety of technical domains, resulting in the creation of numerous techniques.

Currently, mainstream post-hoc detection methods can be further categorized into zero shot detection and fine-tuned classifiers. Rather than relying on additional training data, zero shot detection directly analyses the text using pre-trained language models and classifies it using statistical metrics such as linguistic probability patterns or sensitivity to perturbation. A range of scoring metrics based on probability and similarity have been adopted as zero-shot metrics for detecting machine-generated content. These metrics include:

- Log-Likelihood: The probability of measuring the frequency of the given text in the context of the pre-training language model is indicative of the log likelihood score in the content of the text. The higher the log likelihood score, the more likely it is that the content is generated by an artificial neural network.
- BART Score and BERT Similarity: The utilization of an encoder and a decoder in conjunction with bidirectional transformer models facilitates the identification of semantic similarity between input and reference texts.
- BLEU, ROUGE-L, and TF-IDF Score: The initial NLP indicators were developed for the purpose of machine translation and abstracting the content. In this study, the indicators are employed to quantitatively analyze the content and its variations across different versions.
- Sentence similarity (e.g. cosine similarity in embedding space) and editorial distance: Assessing text tolerance to minor changes - AI-generated content tends to exhibit low variability when rewritten or paraphrased.
- Readability Score: AI-generated texts may exhibit more uniform sentence structure and word usage than human-written texts, resulting in distinguishable readability patterns.

The utilization of these metrics in the context of black-box testing does not necessitate the access to the internal parameters or training data of the model. Consequently, this approach facilitates the provision of a comprehensible, efficient, and independent solution from the model. However, the efficacy of these models is contingent upon the existence of high-quality generation models or extensive editing of the data.

Recent studies have proposed more sophisticated zero-shot techniques. Zhu et al. [12] observed that AI-generated text undergoes less change when rewritten by ChatGPT compared to text written by humans. In light of these observations, a black-box detection method was proposed that does not require access to model parameters. It has been shown that this approach works better than conventional methods on a variety of datasets. However, when managing short texts or restricted API access, this method may have cost and accuracy limitations. To detect model-generated content, Mitchell et al. (DetectGPT) [13] introduced the idea of

"perturbation curvature," which is the difference in log-likelihood between the original text and its mildly perturbed variants. This method is compatible with different pretrained language models and does not require classifier training. However, its resilience to substantial modifications, such as paraphrasing, and to multilingual content, is limited. Gehrmann et al. (GLTR) [14] put forth a methodology that makes use of entropy measures obtained from token frequency rankings in language models. By analyzing the proportion of high-probability tokens in a given text, it is possible to assess the likelihood that it was AI-generated. This method is intuitive and can assist human judgment in certain scenarios. However, its performance degrades when dealing with high-quality generated text or content that has been heavily rewritten. Yang et al. (DNA-GPT) [15] proposed a technique that divides a text into two parts. Subsequently, a language model is employed to predict the latter portion based on the former. The resulting prediction is then compared to the actual text using n-gram overlap. This approach is well-suited for black-box settings and exhibits independence from the model's architecture. However, it tends to be computationally costly and loses effectiveness on short texts or those with a lot of structural variation.

Although that zero-shot detection has drawbacks despite its many benefits, which include strong generalizability, ease of deployment, and the removal of the need for training.

- Dependence on sensitive model outputs, such as log-probability, which are not accessible through certain API-based models.
- The accuracy of detection can be considerably reduced by even small rewordings, making it susceptible to paraphrasing attacks.
- Limited effectiveness on short texts, where insufficient signal is available for reliable detection.
- High computational cost, making it unsuitable for large-scale, real-time detection scenarios.

Constructing a labeled dataset comprising both artificial intelligence-generated and human-written text enables training a fine-tuning classification model. The mainstream architecture paradigm of current AI text detection (AITD) fine-tuning classification models typically uses a feature-based binary classification pipeline. In this pipeline, the input text is first encoded by a text detector (e.g., a transformer-based encoder such as BERT [16], RoBERTa [17], GPT2[18], or T5[19], or a traditional neural architecture such as CNN-LSTM [20 - 24]) to obtain high-level semantic representations. These representations are then fed into a multilayer perceptron (MLP) classifier for binary classification, which determines whether the input text is human- or AI-generated (see Figure 2).

For example, Gifu et al. [25] proposed a two-stage classifier architecture based on Transformer models, achieving detection accuracies of 83% in monolingual and 72% in multilingual scenarios. However, the model demonstrated weak generalization, with performance exhibiting significant influence from the topic and style of the training corpus. Campino [26] concentrated on educational settings by constructing a corpus of academic summaries and fine-tuning BERT to identify student utilization of GPT-generated content. The model enhanced the identification rate of academic dishonesty through the optimization of input processing workflows. However, its application to content generated by other models, such as Bard, has proven unsuccessful, giving rise to concerns regarding transferability and ethical implications. In their seminal work, Wu et al. (LLMDet) [27] pioneered the construction of frequency dictionaries of n-grams generated by large models. These dictionaries represented

a major breakthrough in the field by acting as a stand-in for perplexity benchmarks. These dictionaries were subsequently employed to train classifiers that enhance the detection of black-box model outputs. Although this method has worked well, it is expensive in terms of the resources needed to create and update the dictionaries and is not very reliable when it comes to highly diverse text or stylistic rewrites. Mireshghallah et al. [28] found that smaller or partially trained models tend to be more sensitive to AI-generated text, making them suitable as lightweight detectors. However, the generalization and robustness of these models remain unstable, which limits their applicability across broader contexts.

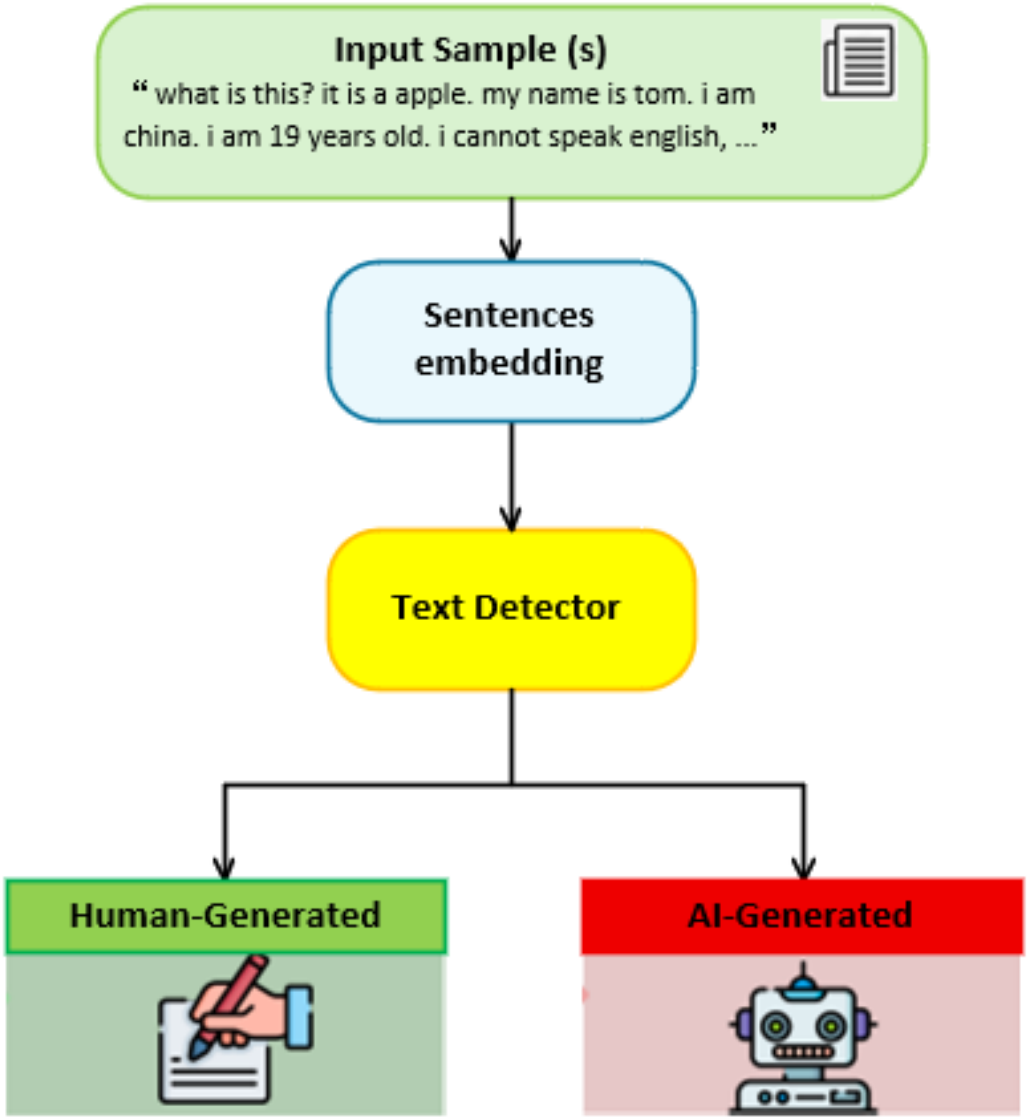


Figure2: Mainstream architecture paradigm of current AI text detection (AITD)

Supervised classifiers have a number of common drawbacks, even though they can achieve high detection accuracy within particular domains or data distributions, making them ideal for focused, closed-loop applications:

- Inherent limitations of relying on deep learning-based feature extraction
- Overfitting to training data, resulting in poor cross-domain generalization.
- Lack of robustness against paraphrasing and text perturbations.
- High cost of constructing training datasets, especially in multilingual and stylistically diverse settings, making them difficult to maintain.
- Continuous maintenance required, as models must be regularly updated to adapt to newly released generative models, incurring substantial upkeep costs.

This study presents a new detection framework that uses deep semantic representations and behavior-based difference features to distinguish AI-generated text from human-written content. Specifically, we use a transformer-based encoder, such as RoBERTa, to extract semantic features from the input text. We also introduce a bi-textual difference feature module to capture the revision features of the AI model. We prompt the language model to revise the input text and then compare the original and revised versions to extract a set of revision-sensitive features, including lexical change ratio, syntactic structure change, surface tagging differences, and multi-granularity similarity metrics. We use a pre-trained semantic coder (e.g., Paraphrase-MiniLM-L6-v2 [29]) for this process. These features are then fused in the Dual Stream Feature Fusion architecture, which learns from both the raw semantic content and the revision-based differences. This approach achieves greater robustness and interpretability in AI text detection tasks.

3. Propsed Methodology

3.1 AI text detection statement

Given two text samples, the objective is to develop a binary classification model that can accurately identify which text is generated by AI and which is human-authored. This classification problem can be mathematically formalized as follows: Let x_i represent a text sample, where i denotes the index of the sample in the dataset. The objective is to categorize the text with a specific label y_i to each x_i , where:

$$y_i = \begin{cases} 0 \ if \ x_i \ is \ human \ generated \\ 1 \ if \ x_i \ is \ AI \ generated \end{cases} \tag{3-1}$$

The problem then becomes one of estimating a function $f : X \to Y$, where X is the space of all possible text samples and $Y = \{0, 1\}$ is the set of possible labels.

Table 3-1: Detailed statistics for OpenGPTText dataset

Subset	OpenGPTText	OpenWebText	Failed to Rephrase	Percentage
urlsf_00	3,888	391, 590	27	0.99%
urlsf_01	3,923	392, 347	0	0%
urlsf_02	3.260	391, 297	652	0.53%
urlsf_03	3, 891	390, 161	10	1.00%
urlsf_04	3, 684	390, 250	218	0.94%
urlsf_05	3, 602	389, 874	296	0.92%
urlsf_06	3, 494	390, 339	409	0.90%
urlsf_09	3, 653	389, 634	243	0.94%
Total	29, 395	3, 125, 469	1, 885	0.94%

3.2 Dataset Description

The OpenGPTText dataset [30] consists of paraphrased textual samples that were generated by the gpt-3.5-turbo language model using the OpenWebText corpus as its source. The data set contains 29,395 textual samples, each corresponding to a piece human-written text from the OpenWebText corpus that shares a same unique identifiifier (UID). The OpenGPTText only contains approximately 1% of paraphrased samples of the OpenWebText data set in some specifific subsets. The number of samples in each subset is listed in table 3-1.

The OpenWebText dataset is a publicly available resource that includes web content from URLs shared on Reddit. The dataset is a reorganization of the original WebText corpus. Since the dataset was compiled in 2019, it is unlikely that the textual content contained therein was generated algorithmically.

3.3 Feature Extraction Based on Dual-Text Differential

We propose a novel Dual-Text Differential Feature Extraction Framework to effectively distinguish AI-generated texts. This framework leverages the revision behavior of AI language models. The key idea is that when prompted to revise their generated text, AI models exhibit consistent and measurable modification patterns. These self-revision behaviors provide a unique signal for detecting synthetic content.

As illustrated in Figure 3, our framework is a multi-stage process that starts with generating AI revisions and ends with extracting rich differential features that quantify semantic and surface-level discrepancies between the original and revised texts.

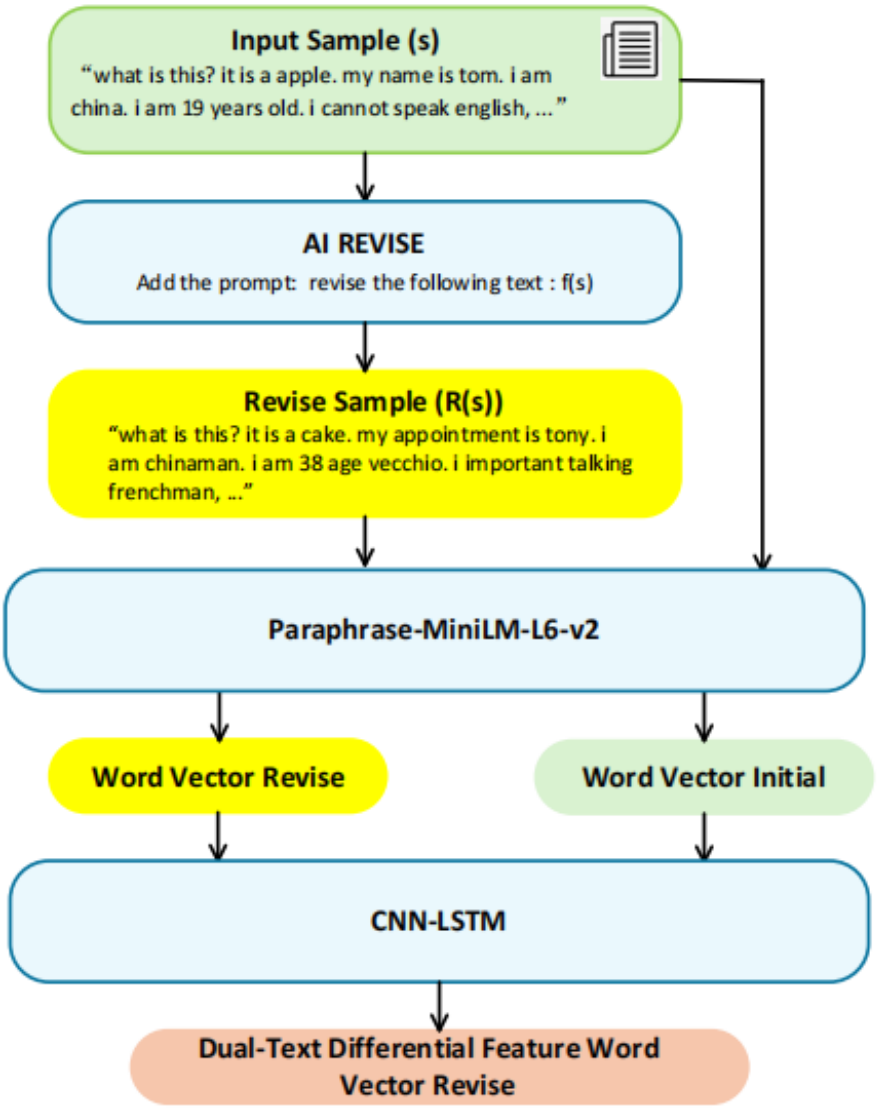


Figure3: Dual-Text Differential Feature Extraction Framework

3.3.1 AI Self-Revision Behavior and Feature Design Motivation

A key component of our proposed framework lies in the behavioral asymmetry exhibited by large language models (LLMs) when revising texts of different origins. Specifically, we observe that LLMs revise AI-generated and human-written texts in qualitatively distinct ways. This behavioral divergence serves as a foundational motivation for our differential feature

extraction strategy.

We hypothesize that an LLM's self-revision behavior reflects its internal confidence and familiarity with the input distribution. When the model is asked to revise a passage, its revision intensity—i.e., the degree and type of changes it makes—varies significantly depending on whether the input text was: originally generated by the same or similar model (in-distribution), or written by a human (out-of-distribution).

This leads to a measurable and systematic difference in the revision footprint, which we aim to capture through a set of differential features.

Empirically, we find that [13]:

AI-generated texts, when revised by an LLM, tend to receive minimal modifications, typically limited to:

- Synonym replacements (e.g., “thus” → “therefore”);
- Minor fluency improvements;
- Punctuation or stylistic adjustments.

Human-written texts, when revised by the same LLM, often undergo more substantial transformations, including:

- Structural reorganization of sentences;
- Standardization of lexical expressions;
- Rewriting of informal or idiosyncratic phrasing.

This difference can be attributed to the fact that AI-generated texts inherently align with the model's learned distribution, while human-authored texts often deviate from it in terms of phrasing, structure, or word choice.

Based on this insight, we design a suite of revision-sensitive features derived from the comparison between an original text s and its AI-generated revision $R(s)$. These features aim to capture the revision intensity and nature, and Semantic Shift: Cosine similarity between sentence embeddings (e.g., from paraphrase-MiniLM-L6-v2); Lexical Modification Metrics: Vocabulary change ratio, token insertion/deletion rates; Syntactic Variation: Sentence count ratio, clause restructuring indicators; Surface-Level Changes: Edit distance, punctuation/capitalization modifications, character-level overlap. These features are then aggregated into a behavior-informed representation, which encodes the LLM's revision behavior as a discriminative signal for downstream classification tasks.

In summary, we exploit a novel source of behavioral information—AI self-revision dynamics—to distinguish between machine-generated and human-written texts. Our framework operationalizes this insight by eliciting revisions from the model and quantifying the behavioral differences through structured feature engineering. This revision-aware design significantly enhances the interpretability and generalizability of the detection model.

3.3.2 Sentence Embedding and Semantic Similarity Modeling

In order to portray the semantic changes between the original text and its corresponding revised version, we have devised a method for modeling semantic differences based on pre-trained sentence encoding models. The method aims to capture the semantic shifts presented in the input text after it has been revised by an AI language model, to provide discriminative feature representations for downstream tasks.

Sentence embedding is a critical step in many NLP tasks, as it transforms a variable-length sentence into a fixed-size vector that captures its semantic meaning. Existing sentence

embedding approaches can be broadly categorized as follows:

- Average Word Embeddings: Compute word vectors using pretrained word embeddings (e.g., GloVe, word2vec), then average them to obtain the sentence embedding. The method is simple, fast, and computationally cheap. But it ignores word order and syntactic structure; lacks contextual information.
- Recurrent/Convolutional Encoders: Use RNNs (e.g., LSTM, GRU) or CNNs to encode the sequence of word embeddings into a sentence representation. It can capture local and sequential information but require supervised training; slow to compute; underperform compared to transformers in many tasks.
- Transformer-Based Sentence Encoders: With the advent of BERT and other transformer models, sentence embedding techniques have shifted toward contextualized embeddings. However, standard BERT models are not optimized for sentence-level similarity, and their [CLS] token embeddings may not reflect semantic closeness directly.

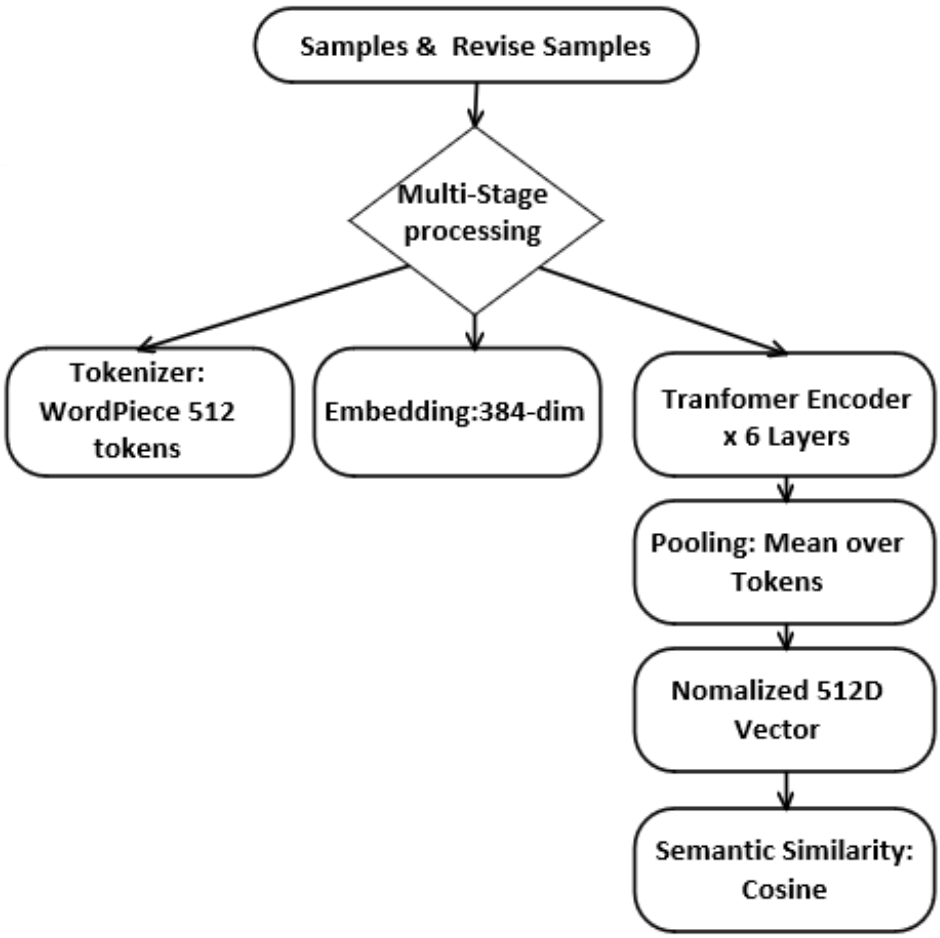


Figure 4: The Framework of paraphrase-MiniLM-L6-v2

3.3.3 Semantic Distance Modeling with paraphrase-MiniLM-L6-v2

In our framework, we use the Paraphrase-MiniLM-L6-v2 model as the sentence embedding encoder, as shown in Figure 4. Developed and released by the Sentence-Transformers library, this model is a distilled version of the MiniLM Transformer architecture.

It comprises six Transformer layers and is optimized for semantic textual similarity and paraphrase identification tasks.

Unlike traditional, large-scale language models, such as BERT or RoBERTa, Paraphrase-MiniLM-L6-v2 is lightweight and computationally efficient. It produces 384-dimensional dense embeddings for each sentence. These embeddings capture rich semantic representations while ensuring scalability for high-throughput processing. This is advantageous when handling large-scale revision pairs in our differential feature extraction pipeline.

We fine-tune the model using a combination of Natural Language Inference (NLI) and Semantic Textual Similarity (STS) datasets. This training strategy uses contrastive learning objectives to teach the encoder to bring semantically similar sentences closer together in the embedding space and push dissimilar ones apart. This training strategy enables the model to capture subtle variations in meaning, such as those found in revision behaviors induced by AI rewriting prompts.

We chose paraphrase-MiniLM-L6-v2 for several reasons. First, it is efficient, allowing for low-latency inference and seamless integration into real-time or batch systems. Second, although paraphrase-MiniLM-L6-v2 is small compared to larger models such as BERT-base (110 million parameters), its performance on benchmark STS tasks is competitive. Most importantly, the model excels at detecting paraphrase-level transformations, which is crucial for our use case. Often, the differences between the original and revised texts are subtle and localized, especially in the case of AI-generated self-revisions.

Using paraphrase-MiniLM-L6-v2 to encode both the original and revised texts allows us to compute fine-grained semantic distances and similarity metrics. These behavioral signals comprise our Revision-Sensitive Feature Set, which enhances the discriminative power of our downstream classifier when distinguishing between human-written and AI-generated content.

3.3.4 Dual-Text Differential Features

To capture the nuanced semantic shifts induced by AI-driven revisions, we propose a feature extraction strategy called Dual-Text Differential Features. This method uses sentence-level semantic embeddings from the paraphrase-MiniLM-L6-v2 model to encode the subtle differences between the original and revised versions of a text pair.

Concretely, for each text pair (T_{origin}, T_{revise}) , we first compute their respective sentence embeddings $V_{origin}, V_{revise} \in \Re^{384}$ using the paraphrase-MiniLM-L6-v2 encoder. These two vectors are then concatenated into a matrix $V = [V_{origin}; V_{revise}] \in \Re^{2\times 384}$, forming a two-channel representation of the sentence pair.

To capture local interactions and sequential patterns in the embedding space, we apply a Convolutional Neural Network (CNN) to extract n-gram level difference features, followed by a Long Short-Term Memory (LSTM) layer to model long-range dependencies and semantic flow across embedding dimensions. As shown in Figure 5, this CNN-LSTM module transforms the original concatenated embeddings into a refined 384-dimensional differential feature vector that encodes not only the magnitude but also the structural dynamics of semantic change.

This enhanced difference vector is a core component of our Dual-Text Differential Features,

providing a deeper context-aware representation of revision behavior. Empirically, we find that CNN-LSTM augmented difference features outperform raw difference vectors in downstream tasks, highlighting the importance of structured modelling for semantic change.

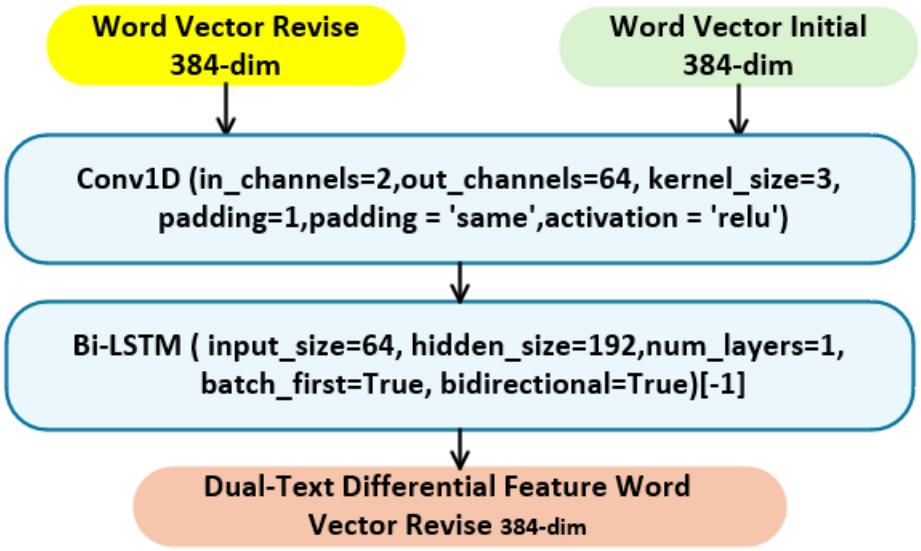


Figure 5: CNN-LSTM Models Used for Concatenating Embeddings

3.4 AIGTD Models Based on Dual-Stream Feature Fusion framework

To effectively distinguish between human- and AI-generated texts, we propose an AI-Generated Text Detection (AIGTD) model based on a Dual-Stream Feature Fusion framework. This framework uses both the original input and the AI-revised version to detect subtle semantic differences that indicate machine-generated content as illustrated in Figure 6.

Specifically, given an input sample s, an instruction-tuned language model is prompted to produce a revised version $R(s)$ through a controlled generation process. Both s and $R(s)$ are then encoded using the Paraphrase-MiniLM-L6-v2 model to obtain dense semantic embeddings. The architecture comprises two specialized CNN-LSTM branches:

- Differential Branch: The first CNN-LSTM stream receives the semantic difference vector, computed as the element-wise absolute difference between the embeddings of s and $R(s)$. This branch is designed to capture fine-grained semantic changes, reflecting revision-aware signals indicative of AI-generated alterations.
- Dual-Sentence Branch: The second CNN-LSTM stream jointly processes the original sentence and its revised version as parallel inputs. This branch aims to extract deep semantic representations that preserve the essential content characteristics of both texts and enhance contextual comprehension.

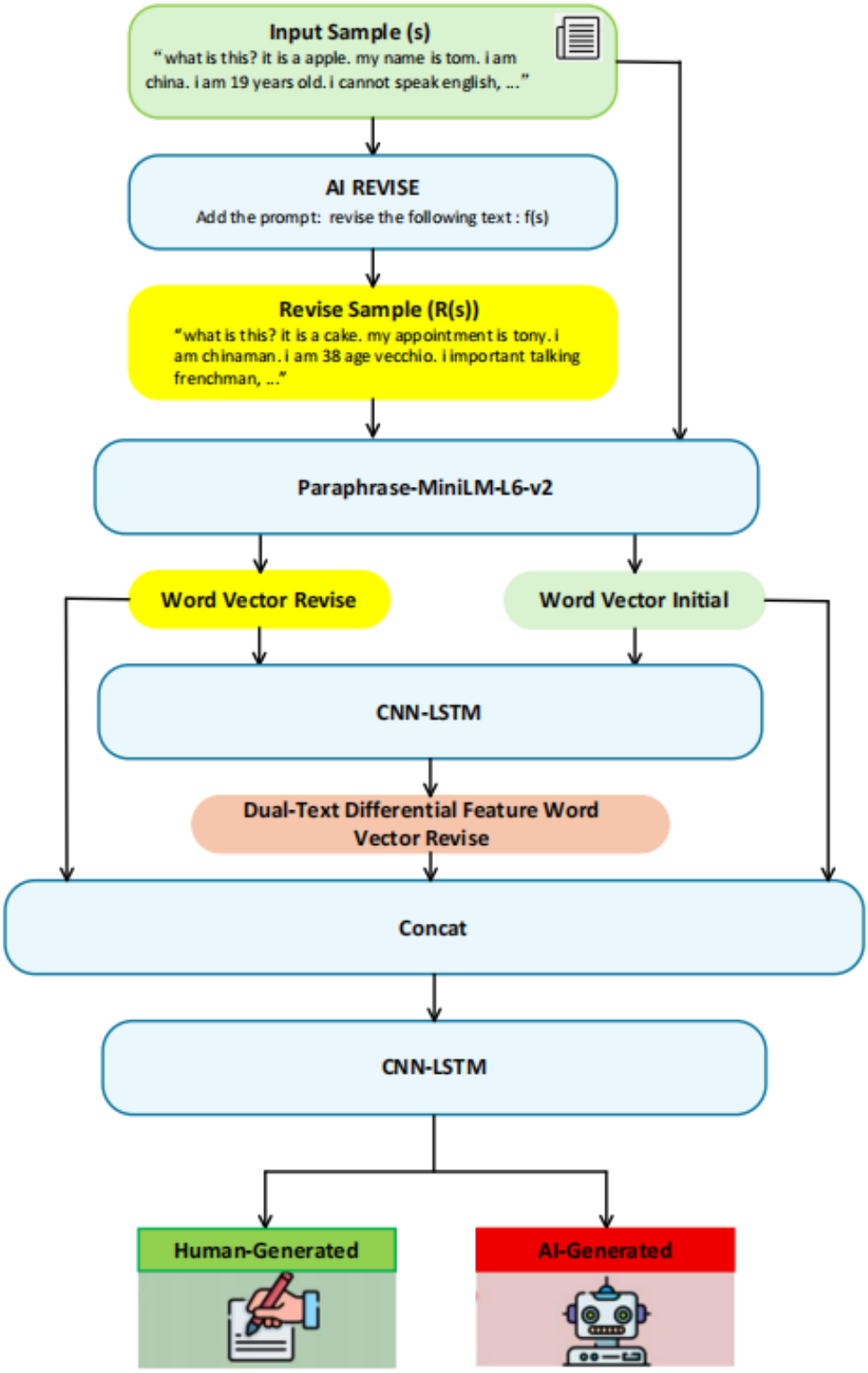


Figure 6: AIGTD Models Based on Dual-Stream Feature Fusion framework

The outputs from both branches are concatenated to form a unified feature representation. This representation is then passed through a classification layer to determine whether the text is human-written or machine-generated.

The proposed dual-stream architecture explicitly models both the semantic essence and revision-induced discrepancies. This provides a robust and interpretable mechanism for

detecting AI-generated text, particularly in scenarios involving subtle paraphrastic transformations.

4. Experiments

4.1 Evaluation Metric

Evaluating the model's performance is a crucial aspect of this study, encompassing assessments for both non-fine-tuned models and at each epoch during the fine tuning process. This comprehensive approach facilitates the computation of average performance metrics over multiple epochs.

In this paper, the term "positive" refers to the input text is ChatGPT-generated, while "negative" , means that the data is written by human.

Given the true positive (T_p), true negative (T_N), false positive (F_p) and false negative (F_N) count, we can calculate the metrics as following:

Firstly, accuracy is the proportion of correctly classified examples out of all examples. It is computed as the number of true positives and true negatives divided by the total number of examples as per Eq. (4-1).

$$\text{accucary} = \frac{T_p + T_N}{T_p + T_N + F_p + F_N} \tag{4-1}$$

Secondly, precision is the proportion of true positives among all examples classified as positive. It is computed as the number of true positives divided by the total number of positive predictions as per Eq. (4-2).

$$precision = \frac{T_p}{T_p + F_p} \tag{4-2}$$

Thirdly, recall is the proportion of true positives among all actual positive examples. It is computed as the number of true positives divided by the total number of positive examples as per Eq. (4-3).

$$recall = \frac{T_P}{T_P + F_N} \tag{4-3}$$

Fourthly, F1-score is the mean of precision and recall. It is a measure of the balance between these two measures as per Eq. (4-4).

$$F_1 \text{ score} = \frac{2 \times presision + recall}{presision + recall} \tag{4-4}$$

Fifthly, AUC-ROC is the area under the receiver operating characteristic (ROC) curve. It is a measure of the model’s ability to distinguish between positive and negative examples.

The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for different threshold values. The AUC-ROC is computed as the area under the ROC curve. Lastly, AUC-PR: AUC-PR is the area under the precision-recall curve. It is a measure of the model’s ability to retrieve positive examples. The precision-recall curve plots the precision against the recall for different threshold values. The AUC-PR is computed as the area under the precision-recall curve.

4.2 Experimental Results and Analysis

4.2.1 Data set partitioning and fine-tuning strategy

To ensure experimental rigor and validate the effectiveness of our approach, we employ a specific dataset partitioning and model fine-tuning strategy. Two independent OpenGPTText datasets, both related to the financial domain, are used in this study.

The first dataset is dedicated to model fine-tuning. We use this dataset to fine-tune the Paraphrase-MiniLM-L6-v2 model, with the core goal of enhancing the model's semantic representational ability in distinguishing between human text and AI text, so that it can capture the semantic similarity differences between the two more acutely.

The second financial dataset serves as the main testbed for all our comparison experiments. For model training and evaluation, we strictly divide this dataset into a 70% training set and a 30% testing set. All the experimental results shown in the subsequent sections, including performance comparisons of different feature types and model architectures, are done on this uniformly partitioned dataset, which ensures the fairness and reproducibility of the results of all the comparison experiments.

4.2.2 Validity analysis

To comprehensively evaluate the effectiveness of the proposed approach, we designed a set of experiments from two perspectives based on OpenGPTText dataset: the discriminative power of the Dual-Text Differential features, and the validity of the proposed Dual-Stream Feature Fusion framework. We adopt multiple evaluation metrics including Accuracy, Precision, Recall, F1-score, and AUC.

(1) Effectiveness of Dual-Text Differential Features

The first line of experimentation is centered on validating the standalone effectiveness of the Dual-Text Differential Features, which are derived from the semantic differences between an input sentence and its model-generated revision. In the first set of experiments, we investigate the impact of using only original semantic features versus only differential features across four popular pre-trained language models: BERT, RoBERTa, paraphrase-MiniLM-L6-v2, and GPT-2. For each encoder, we evaluate two independent feature configurations:

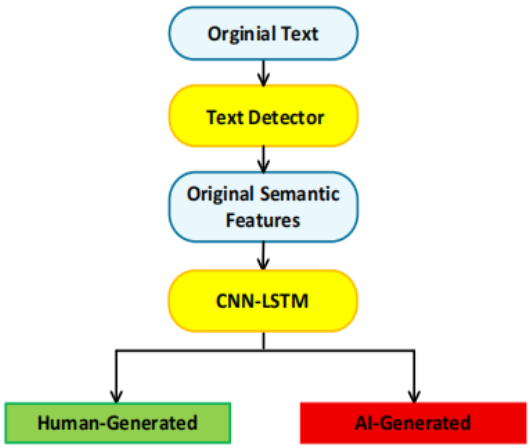
- Original Semantic Features Only (As shown in Figure 7(a)): sentence embeddings extracted directly from the original text without incorporating revision information.
- Dual-Text Differential Features Only (As shown in Figure 7(b)): sentence-level difference vectors computed as the absolute value between the embeddings of the original and revised sentences.

By isolating these two feature types, we aim to quantify the specific contribution of

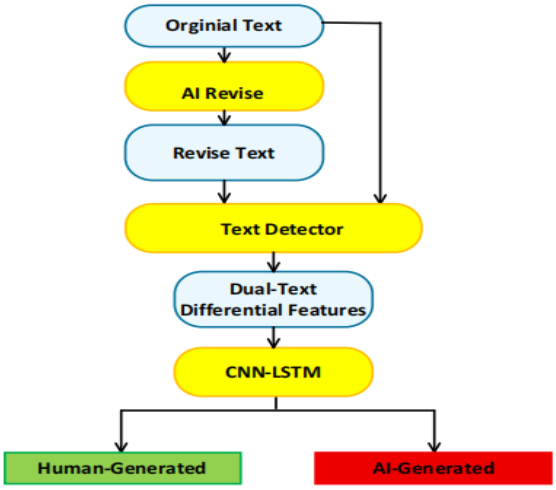
semantic difference information to the downstream AI-generated text detection task, and to determine whether such differential signals offer added value across different pre-trained language models.

As shown in Table 4-1, Table 4-2, and Figure 8 - 9, the results show that: For BERT, RoBERTa, and paraphrase-MiniLM-L6-v2, models trained solely on Dual-Text Differential Features consistently outperform those trained only on original semantic features across all evaluation metrics. This indicates that semantic differences derived from AI-driven revision behavior provide valuable discriminative signals for AI-generated text detection (AIGTD).

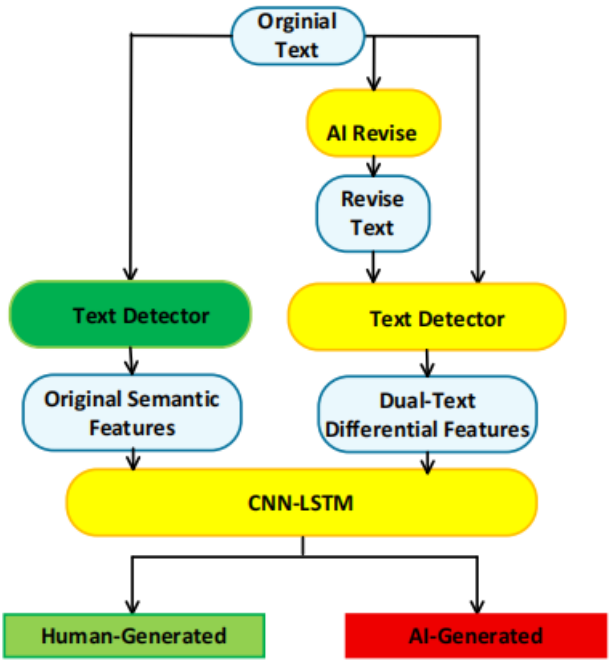
These findings confirm the general effectiveness and robustness of the proposed Dual-Text Differential feature extraction mechanism, which captures nuanced semantic shifts resulting from the model's self-revision process.



(a) Original Semantic Features Only



(b) Dual-Text Differential Features Only



(c) Original Semantic Features and Dual-Text Differential Features

Figure 7: Three models for the comparison experiment.

(2) Effectiveness of the Dual-Stream Feature Fusion Framework

The second axis of experimentation is designed to evaluate the overall efficacy of the Dual-Stream Feature Fusion Framework proposed in this study. This architecture jointly leverages both the original semantic features and the Dual-Text Differential Features through two dedicated CNN-LSTM branches, as detailed in Section 3.4.

To assess the benefits of such architectural fusion, we compare the performance of the Dual-Stream model with single-branch baselines (As shown in Figure 7c), where only one type of feature (either semantic or differential) is used for downstream classification. This enables us to test the hypothesis that combining intrinsic textual semantics with model-induced revision discrepancies leads to more accurate and robust detection of AI-generated content.

Together, these two dimensions of experimentation provide a comprehensive evaluation of both the feature-level contributions and the architectural advantages embedded in our proposed method.

As shown in Tables 4-1 to 4-3 and related figures, our experimental results strongly demonstrate the superior effectiveness of our proposed feature engineering strategy and Dual-Stream Feature fusion framework in the AI-generated text detection task. By comprehensively evaluating multiple encoder backbones, we draw the following core conclusions:

First, the Dual-Stream Feature Fusion Framework (DSFF) consistently outperforms any single-feature model. As shown in Table 4-3, almost all performance metrics (accuracy, precision, recall, F1 score, and AUC) peak for all models when both raw semantic features and bi-textual difference features are used. This conclusion is intuitively and strongly supported in our visualization charts. As can be clearly seen from the comparison graph of ROC curves (shown in Figure 8a), all the models that have undergone dual-stream fusion (BERT, DistilBERT, RoBERTa, Fine-tune model) reach a perfect AUC value of 1.000, and their curves perfectly fit the upper-left boundary, which represents the achievement of the desired classification, far exceeding the performance of the single-feature model. Similarly, the precision-recall curves (shown in Figure 8b) show a similar trend, with the curves of the fused models also fitting the perfect region in the upper left corner. Among all base models, RoBERTa shows the strongest performance when using original semantic features, achieving the highest Accuracy (99.87%), suggesting it is most effective at capturing static semantic content.

In addition, the confusion matrix (shown in Figure 8c) further reveals the strong discriminative power of the DSFF framework. In the case of the RoBERTa model, the number of false positives (misclassifying humans texts as AIs) and false negatives (misclassifying AIs texts as humans) are only 8 and 1, respectively, with an extremely low total number of classification errors. The classification metrics bar chart (shown in Figure 8d) also quantifies this advantage, with all fusion models approaching a perfect score of 1.0 for precision, recall, and F1 scores. Together, this visual evidence demonstrates that there is a strong complementarity between the intrinsic static semantic information of the text and the dynamic difference signals generated by the AI revision behavior, and that our DSFF framework successfully exploits this complementary advantage to achieve near-perfect detection performance.

Table 4-1: The evaluation results for different models based on Original Semantic Features Only

Method	GPT-2	T5	BERT	RoBERTa	Paraphrase-MiniLM-L6-v2
AUC	0.3822	0.5570	0.5112	0.2485	0.8944
Accuracy	0.5005	0.5492	0.5175	0.5008	0.8243
Precision	0.75	0.55	0.53	0.75	0.82
Recall	0.5	0.55	0.52	0.5	0.82
F1-socre	0.33	0.54	0.48	0.34	0.82

Table 4-2: The evaluation results for different models based on Dual-Text Differential Features Only

Method	GPT-2	T5	BERT	RoBERTa	Paraphrase-MiniLM-L6-v2	Finetuned Paraphrase
AUC	0.9828	0.9827	0.9836	0.9825	0.9813	0.9987
Accuracy	0.9368	0.9387	0.9377	0.9368	0.9387	0.9863
Precision	0.94	0.94	0.95	0.94	0.94	0.99
Recall	0.94	0.94	0.95	0.94	0.94	0.99
F1-socre	0.9363	0.9385	0.94	0.99	0.9384	0.9865

Table 4-3: The evaluation results for different models based on Original Semantic Features and Dual-Text Differential Features

Method	GPT-2	BERT	RoBERTa	Paraphrase-MiniLM-L6-v2
AUC	0.6966	0.9999	1.0000	0.9998
Accuracy	0.5400	0.9928	0.9962	0.9898
Precision	0.5601	0.9929	0.9962	0.9900
Recall	0.5400	0.9928	0.9962	0.9898
F1-socre	0.5873	0.9928	0.9962	0.9898

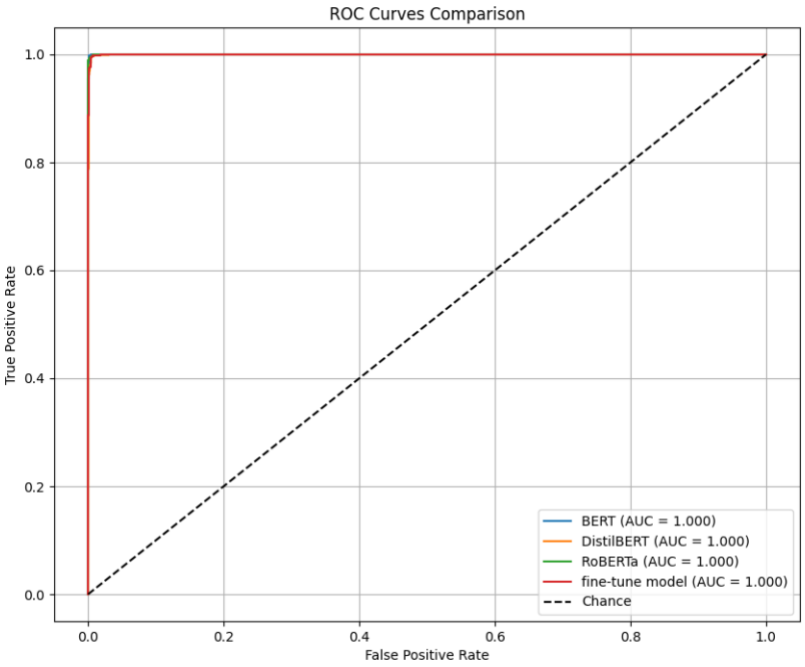


Figure 8(a): ROC for DSFF

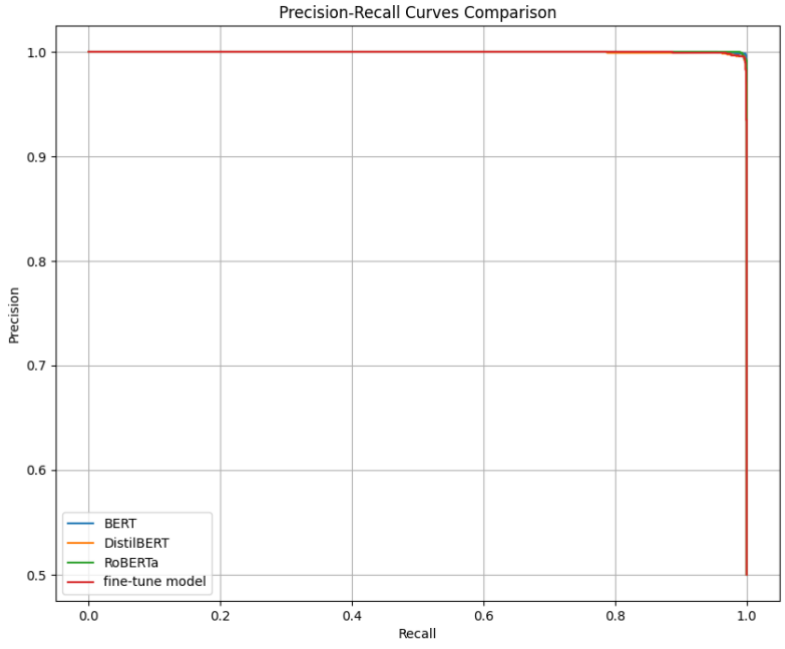


Figure 8(b): Recall and Precision for DSFF

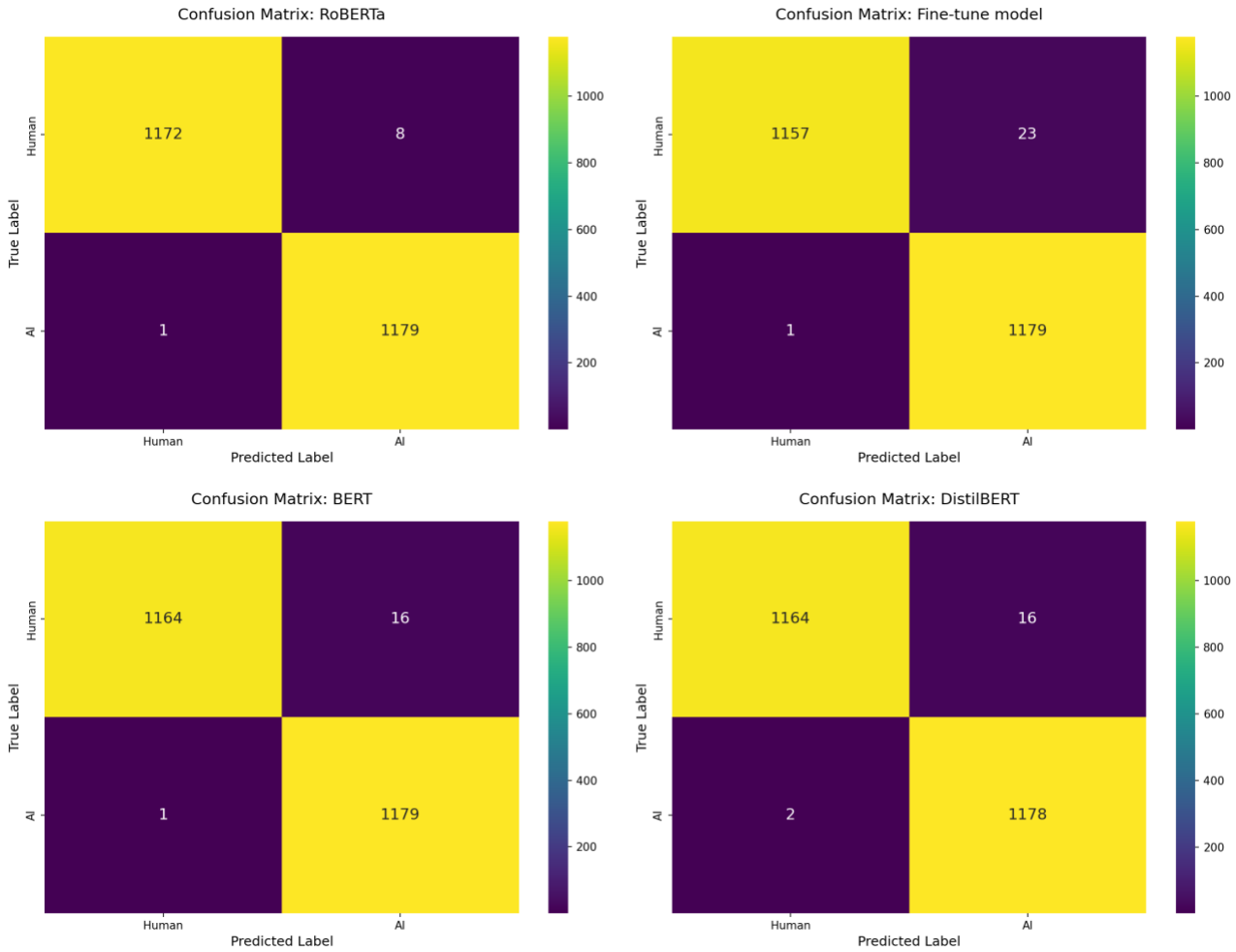


Figure 8(c): Confusion matrix for DSFF

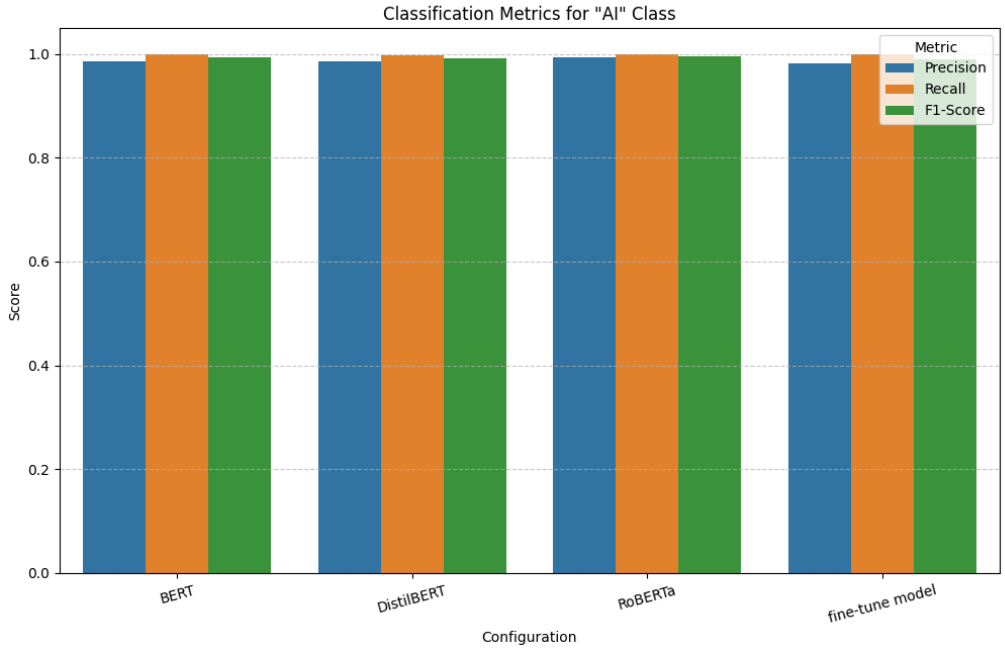


Figure 8(d): Bar chart of disaggregated indicators for DSFF

Second, the effectiveness of feature engineering is well validated, but the sensitivity of different models to different features varies.

- A noteworthy finding in the benchmark tests using dual-text differential features Only (Table 4-2) is that the fine-tuned Paraphrase-MiniLM-L6-v2 model achieves the best performance with an accuracy of 98.63% and an F1 score of 98.65%, comprehensively outperforming all other models including RoBERTa. This finding is further corroborated in our visualization charts:
  - ROC curve analysis (shown in Figure 9a): the fine-tuned model (red line) exhibits the highest area under the curve (AUC = 0.999), and its curve is closest to the upper left corner, suggesting that it has the best combined classification ability at all thresholds.
  - Confusion matrix analysis (shown in Figure 9b): Compared to the standard Paraphrase model, the fine-tuned model performs excellently on the confusion matrix, with the number of false positives (misclassifying humans texts as AIs) and false negatives (misclassifying AIs texts as humans) (19 and 24, respectively) being much lower than that of all the other models (which are generally in the range of 60-130), demonstrating its extremely high discriminative accuracy.
  - Categorical Metrics Comparison (shown in Figure 9c): In the bar chart comparison of precision, recall, and F1 scores, the fine-tuned model achieves the highest scores on all three metrics, once again demonstrating its all-around superiority. Together, this visual evidence suggests that models optimized for specific tasks, such as paraphrase recognition, have a natural and powerful advantage in understanding and characterizing the intrinsic semantic aspects of the original text without the need for additional differential features.

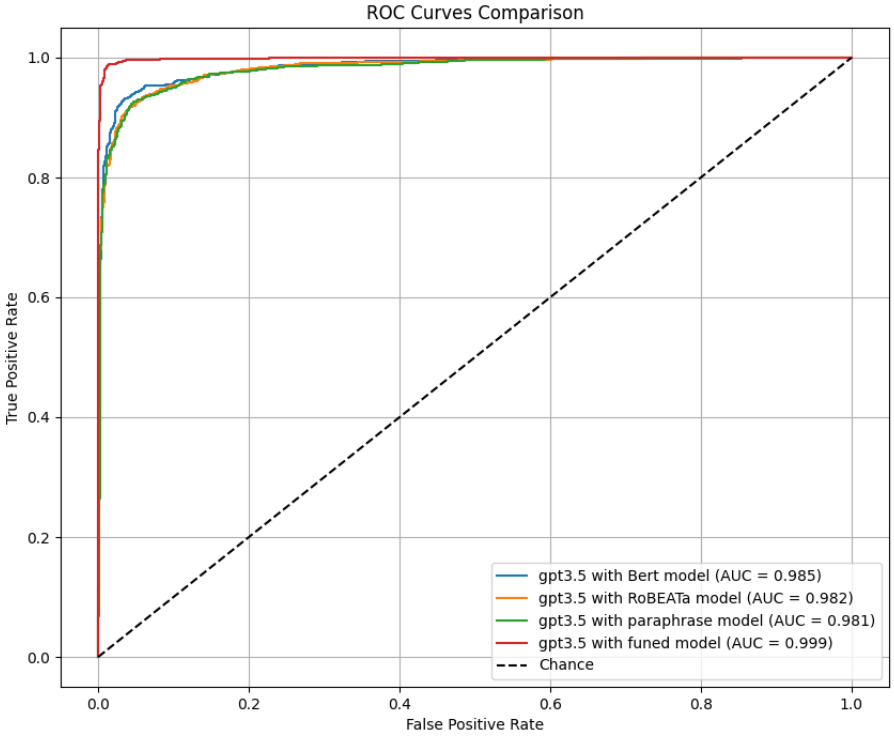
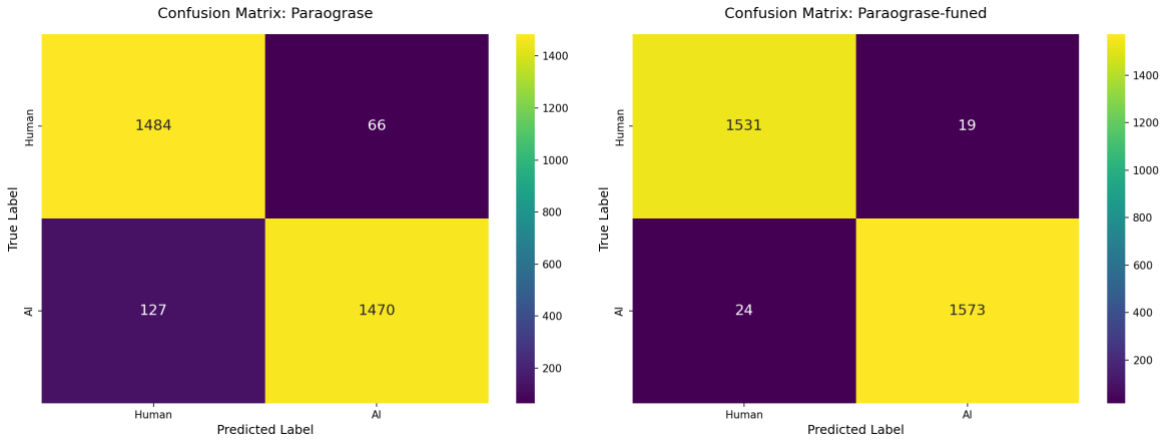


Figure 9(a): ROC for dual-text



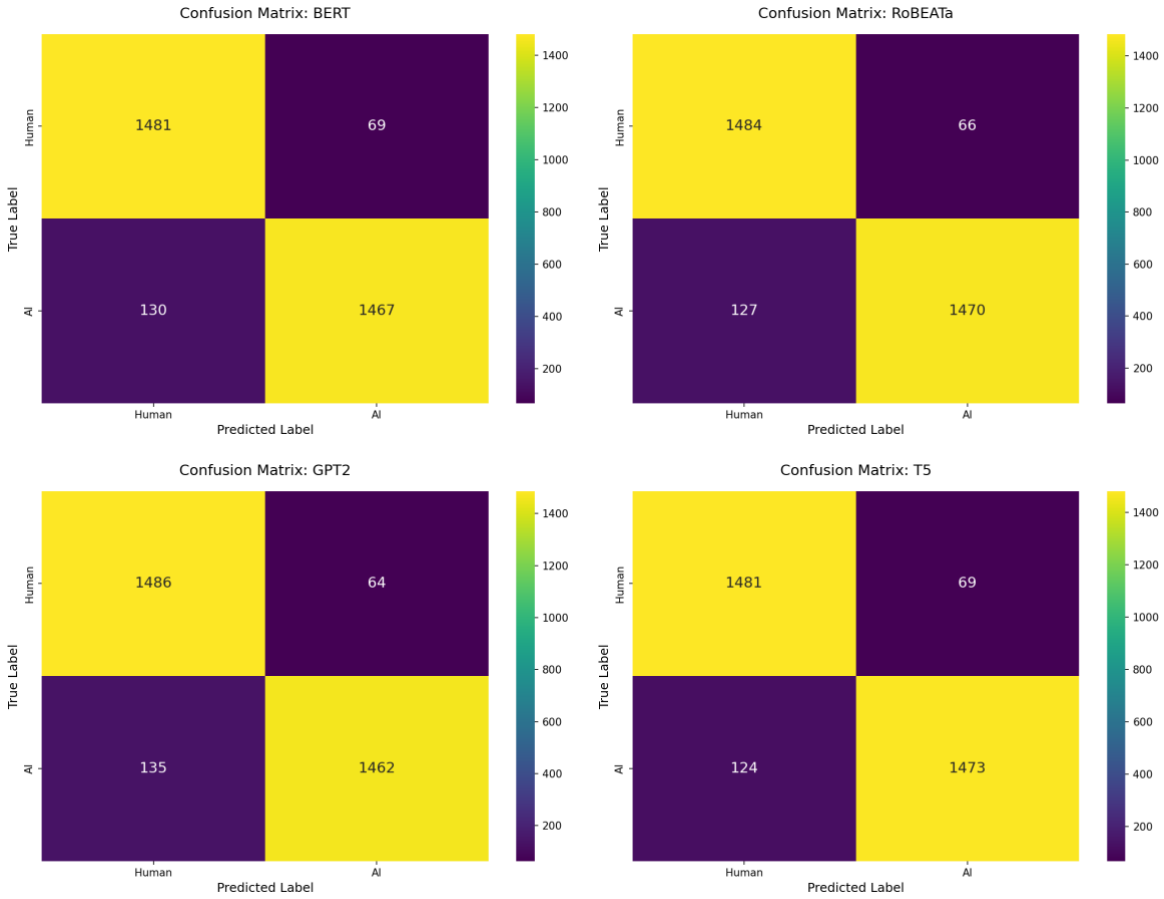


Figure 9(b): Confusion matrix for dual-text

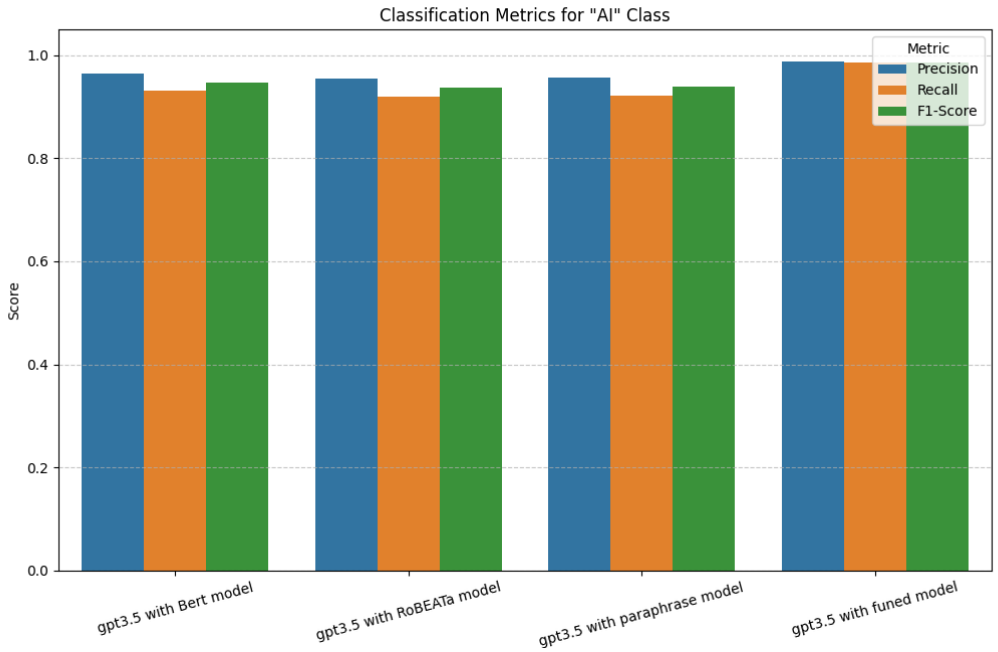


Figure 9(c): Bar chart of disaggregated indicators for dual-text

- However, in the test using original semantic features Only (Table 4-1), we obtain very different results from the powerful performance of the double text difference feature. The experimental results clearly reveal that general-purpose language models (e.g., BERT, RoBERTa) that are not fine-tuned for a specific task perform poorly in identifying the semantic similarity differences between AI and human texts, and even show the opposite judgment pattern to the one expected; whereas the paraphrase-MiniLM-L6-v2 model, which is optimized specifically for the computation of semantic similarity, exhibits excellent performance with an AUC as high as 0.8944, an accuracy of 82.4%, and a classification accuracy of 82.43% at the optimal threshold (0.6184). Its ROC curve (dark purple solid line in Figure_1) is closest to the upper left corner, indicating that it has the strongest comprehensive classification ability under all thresholds. The F1 scores are all around 0.82. This result strongly demonstrates that the performance of a model is not solely dependent on its underlying architecture (e.g., BERT), but also on whether it is effectively optimized for a specific downstream task (e.g., semantic similarity computation). It is demonstrated that task-specific fine-tuning is the key to unlocking the potential of models in niche domains. In contrast to dedicated models, generic models without specific fine-tuning, such as BERT and T5 model, perform mediocrely. BERT has an AUC of only 0.5112, while T5 is 0.5570, both of which are at the random guessing baseline of 0.5 hovering around the neighborhood. Although their classification reports show accuracies of over 50%, this is achieved by choosing an extreme "optimality threshold", whose confusion matrix reveals a serious classification bias. This suggests that directly using the average word vector strategy of a generalized model to generate sentence embeddings is not effective in capturing the subtle semantic differences needed to distinguish between human and AI text. The most thought-provoking findings from this experiment come from RoBERTa and gpt2. both models have AUC values significantly lower than 0.5 (0.2485 for RoBERTa and 0.3822 for gpt2), presenting an inverse categorization capability. We infer that this anomalous behavior stems from the data features and the language preference inherent in the model. Human answers are usually more concise and direct, while AI-generated answers may be more detailed and structured. Models such as RoBERTa and gpt2 may be more inclined to consider concise, centralized answers (human style) as a higher degree of semantic fit to the question when performing semantic matching, whereas detailed answers with additional explanations (AI style) may be judged as less similar due to their "redundancy of information". The results of this experiment also answer the question of why we subsequently used the paraphrase-MiniLM-L6-v2 model as the main fine-tuning model.

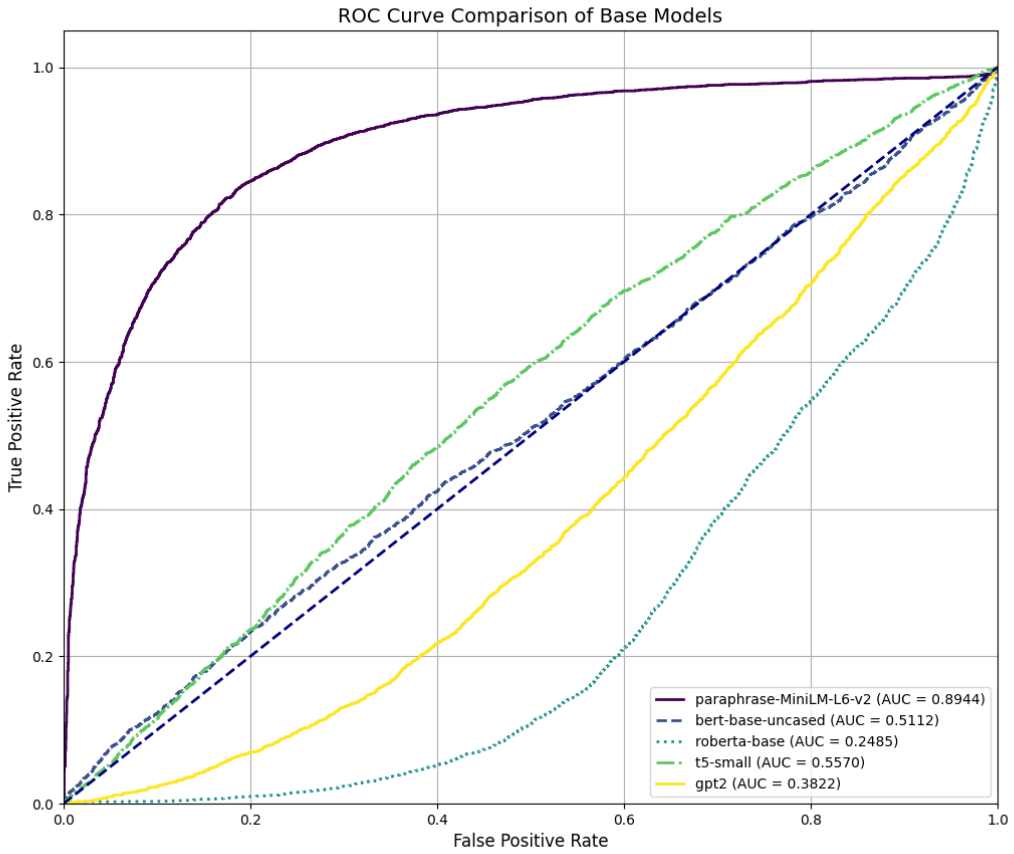


Figure 10: ROC for original semantic features

Finally, the combination of model selection and feature type is key to achieving the best performance. Although fine-tuned Paraphrase model performed best on dual text features and the Paraphrase model led on raw features, it was still the dual-stream feature fusion model constructed on RoBERTa that ultimately achieved the best performance across the board. The model is near-perfect on all metrics (AUC = 1.0000, Accuracy = 0.9962, F1 Score = 0.9962), which is a strong testament to the ability of our DSFF framework to most effectively combine RoBERTa's powerful disparity signaling capabilities with raw semantic information for synergistic efficiency gains.

In conclusion, the experimental results validate the effectiveness and generality of the proposed Dual-Text Differential features in enhancing the performance of AIGTD across multiple backbone models. Moreover, the Dual-Stream Feature Fusion framework substantially improves detection performance by jointly modeling the semantic content and revision-induced differences. These findings highlight the potential of integrating AI revision behaviors into semantic representation learning for robust AI-generated text detection.

5. Conclusion

The core contribution of this research is to propose a novel AI-generated text detection (AIGTD) framework that improves detection accuracy by exploiting the semantic revision behavior of large language models. We introduce the concept of Dual-Text Difference Feature (DTDF), which captures the subtle semantic differences between the original text and its model-generated revision via a deep network, thus enabling the identification of machine-generated content from a unique, behavior-based perspective. In addition, we design the Dual-Stream Feature Fusion Architecture (DSFF) to effectively integrate the static semantic representations of the original text with the dynamic signals of the differences, thus enabling a more comprehensive assessment of the authenticity of the text.

We have conducted extensive experiments on a variety of Transformer-based encoders (e.g., BERT, RoBERTa, GPT-2 and Paraphrase-MiniLM-L6-v2), and the results fully demonstrate the effectiveness and robustness of our approach. The experiments show that in most cases, DTDF alone already has very high discriminative power, which outperforms the baseline model using only raw semantic features in key metrics such as accuracy, F1 score and AUC. More importantly, the DSFF model consistently demonstrated the best performance among all tested backbone models (GPT-2 is the only exception due to its architectural properties), which strongly confirms the common value of combining raw semantic cues with differentiated semantic cues. These findings highlight the great potential of using AI models' own revision behaviors as a source of discriminative features, opening new research directions for building more interpretable and behavior-driven AIGTD strategies.

Although our proposed DSFF framework has achieved excellent results in several backbone models, there are still some limitations in this study that clarify the direction for future modifications and optimizations. First, all the experiments in this study are based on a single AIGTD dataset, and although the validity of our method in the current work scenario is verified, its generalization ability to different language domains, text styles, and outputs of novel generative models still needs to be further evaluated. Second, the feature extraction process relies on an additional LLM API call, which significantly increases the computational cost and time delay. Future work could explore extending this framework to multilingual scenarios, investigating its robustness under adversarial attacks, and exploring more efficient mechanisms for differential feature extraction to ensure its applicability and scalability in real-world applications.

In conclusion, the DSFF framework in this study has very high potential for the AIGTD task. By incorporating AI revision behavior into semantic representation learning, we provide a practical path for more robust and reliable AI-generated text detection.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[4] Chris Stokel-Walker. Ai bot chatgpt writes smart essays-should academics worry? Nature, 2022.

[5] Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. On the possibilities of ai-generated text detection. arXiv preprint arXiv:2304.04736, 2023.

[6] Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. arXiv preprint arXiv:2302.10149, 2023.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[8] Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. arXiv

preprint arXiv:2303.13408, 2023.
[9] Xiaodong Wu, Ran Duan, and Jianbing Ni. Unveiling security, privacy, and ethical concerns of chatgpt. arXiv preprint arXiv:2307.14192, 2023b.
[10] Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. Cryptology ePrint Archive, 2023a.
[11] Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. arXiv preprint arXiv:2303.13408, 2023.
[12] Zhu B, Yuan L, Cui G, et al. Beat llms at their own game: Zero-shot llm-generated text detection via querying chatgpt[C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023: 7470-7483.
[13] Mitchell, Eric, et al. "Detectgpt: Zero-shot machine-generated text detection using probability curvature." International Conference on Machine Learning. PMLR, 2023.
[14] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. Gltr: Statistical detection and visualization of generated text. arXiv preprint arXiv:1906.04043, 2019.
[15] Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. Dna-gpt: Divergent ngram analysis for training-free detection of gpt-generated text. arXiv preprint arXiv:2305.17359, 2023.
[16] 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, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019.
[17] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv 2019, arXiv:1907.11692.
[18] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffifie, and Jasmine Wang. Release strategies and the social impacts of language models, 2019.
[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unifified text-to-text transformer. CoRR, abs/1910.10683, 2019.
[20] Yadagiri A, Shree L, Parween S, et al. Detecting AI-generated text with pre-trained models using linguistic features[C]//Proceedings of the 21st International Conference on Natural Language Processing (ICON). 2024: 188-196.
[21] Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Comput. 1997, 9, 1735–1780.
[22] Garib, A.; Coffelt, T.A. DETECTing the Anomalies: Exploring Implications of Qualitative Research in Identifying AI-generated Text for AI-assisted Composition Instruction. Comput. Compos. 2024, 73, 102869.
[23] Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. arXiv 2014, arXiv:1409.3215.
[24] Lin, Y.; Ruan, T.; Liu, J.; Wang, H. A Survey on Neural Data-to-Text Generation. IEEE

Trans. Knowl. Data Eng. 2024, 36, 1431–1449.
[25] Gifu D, Silviu-Vasile C. Artificial Intelligence vs. Human: Decoding Text Authenticity with Transformers[J]. Future Internet, 2025, 17(1): 38.
[26] Campino J. Unleashing the transformers: NLP models detect AI writing in education[J]. Journal of Computers in Education, 2024: 1-29.
[27] Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. Llmdet: A large language models detection tool. arXiv preprint arXiv:2305.15004, 2023a.
[28] Fatemehsadat Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. Smaller language models are better black-box machine-generated text detectors. arXiv preprint arXiv:2305.09859, 2023.
[29] Reimers N, Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation[J]. arXiv preprint arXiv:2004.09813, 2020.
[30] Chen, Yutian, et al. "Gpt-sentinel: Distinguishing human and chatgpt generated content." arxiv preprint arxiv:2305.07969 (2023).

Statement on the use of generative AI tools

This report uses AI tools (including DeepL and ChatGPT) for grammar checking and partial translation. The sole purpose of using these tools is to correct grammatical errors and improve the readability and clarity of the original English expression.