knowledge

MDPI

Article

Integrating Retrieval-Augmented Generation with Large Language Model for Robust and Explainable AI Text Detection

Ibtasam Ur Rehman [1], Muhammad Islam [2,*], Muhammad Yousaf Rehman [3] and Basharat Hussain [4]

[1] Department of computing, The Millennium Universal College (TMUC), Islamabad 44000, Pakistan; ibtasam.rehman@tmuc.edu.pk
[2] College of Science and Engineering, James Cook University, Cairns, QLD 4878, Australia
[3] Department of Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK; my.rehman007@gmail.com
[4] Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad 44000, Pakistan; basharat.hussain@isb.nu.edu.pk
* Correspondence: muhammad.islam1@jcu.edu.au

Abstract

Large Language Models (LLMs) have been rapidly evolving lately, resulting in the need for strong, explainable models to detect the difference between human-generated and machine-generated articles. Existing approaches which are mostly based on fine-tuned transformers suffer from several drawbacks such as rapid obsolescence, paraphrasing attacks, and lack of interpretability. To improve their ability to detect, this paper proposes a novel paradigm called Human vs. LLM Identification (HLI) which introduces a Retrieval-Augmented Generation (RAG)-inspired evidence-based detection strategy alongside a fine-tuned transformer classifier. Our core model, DeBERTa-Sentinel, is built on top of a fine-tuned Microsoft DeBERTa-v3-small model, which uses a disentangled attention mechanism to better capture subtle syntactic and stylistic deviations characteristic of AI-generated text. We evaluate our framework on a balanced dataset of 43,456 text samples, curated from the OpenGPTText corpus and covering AI-generated and human-authored content across diverse domains including news, education, and creative text. The experimental results show improved performance over the selected baselines, with our framework achieving an accuracy of 97.53%, precision of 95.89%, recall of 99.34%, and ROC-AUC of 99.53%. In addition, explainability is integrated into our framework through Local Interpretable Model-agnostic Explanations (LIME) analysis, providing token-level insight into classification decisions. This study establishes a benchmark for scalable, explainable AI text detection, with implications for academic integrity, content moderation, and combating misinformation.

Keywords: artificial intelligence; natural language processing; Large Language Models; AI text detection; DeBERTa; Retrieval-Augmented Generation; deep learning; text classification

Check for updates

Academic Editor: Jose María Merigo

Received: 30 June 2026
Revised: 18 August 2026
Accepted: 19 August 2026
Published: 25 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the swift progress of Large Language Models (LLMs) like GPT-4, Claude and LLaMA, NLP has been transformed and their ability to create text that closely mimics human writing is becoming more astounding. This advancement has enabled unprecedented productivity in generating content, teaching, and customer service, but also in generating critical issues for academic integrity, information authenticity and digital trust [1]. LLMs have become readily available via commercial APIs and open-source releases making

text generation with AI extremely easy, which has led to the strong need for detection tools. The implications of underdetection are not limited to academic dishonesty, but also reach into disinformation campaigns, loss of trust for digital media and manipulation of public discourse [2]. The evolution of LLMs to more advanced capabilities has made the construction of reliable, explainable and generalizable detection systems an important research task.

There have been several research paradigms to detect AI-generated text, all of which have shown inherent weaknesses. The first ones were based on statistical analysis and/or token-level probability, and were not particularly successful when extended to more complex models [3]. Then, transformer-based architectures came up, such as models based on pretrained encoders and sophisticated attention mechanisms to identify stylistic deviations [4,5]. Zero-shot detection methods, watermarking approaches, and domain adaptation strategies are some of the new developments, cited in [6,7]. There are, however, three major problems with the current systems. The first is that they can easily go stale as the LLM architectures evolve and the models trained on older versions may not be applicable to the newer ones. Second, they are black-box classifiers, which means they do not give reasoning and hence cannot be trusted and used in high-stakes situations. Third, they can be easily attacked by malicious users and have been found to completely fail to detect non-native speech when a simple paraphrase of the input is used, with accuracy dropping from over 90% to under 30% [8,9].

To overcome these drawbacks, this paper proposes a novel detection paradigm, called Human vs. Large Language Model Identification (HLI), which draws on Retrieval-Augmented Generation (RAG) principles alongside transformer-based classification. Our approach is motivated by a departure from purely parametric models, which rely solely on fine-tuning over a fixed collection of texts, toward a framing in which a fine-tuned DeBERTa-v3-small model could be further enriched with a hand-crafted knowledge base of linguistic patterns and stylistic signatures to support more evidence-based classification decisions; we develop this RAG-inspired framing conceptually and outline it as a direction for future implementation as shown as Figure 1. We curate a balanced dataset of 43,456 text samples from the OpenGPTText corpus, spanning AI-generated and human-written text across diverse domains including news journalism, technical documentation, academic writing, and creative fiction. This framework is designed with explainability in mind, using LIME (Local Interpretable Model-agnostic Explanations) to calculate the importance of each feature and analyzing attention patterns to provide additional interpretive insight. We achieve state-of-the-art results on this dataset with 97.53% accuracy, 95.89% precision, 99.34% recall and 99.53% ROC-AUC, substantially outperforming traditional machine learning and deep learning baselines. Ablation experiments confirm that the pretrained transformer backbone and task-specific fine-tuning are the dominant contributors to this performance. This piece represents a significant advance in the interpretation of AI-generated text identification and is scalable, interpretable, and robust, which has profound implications for content moderation, misinformation detection, and academic integrity.

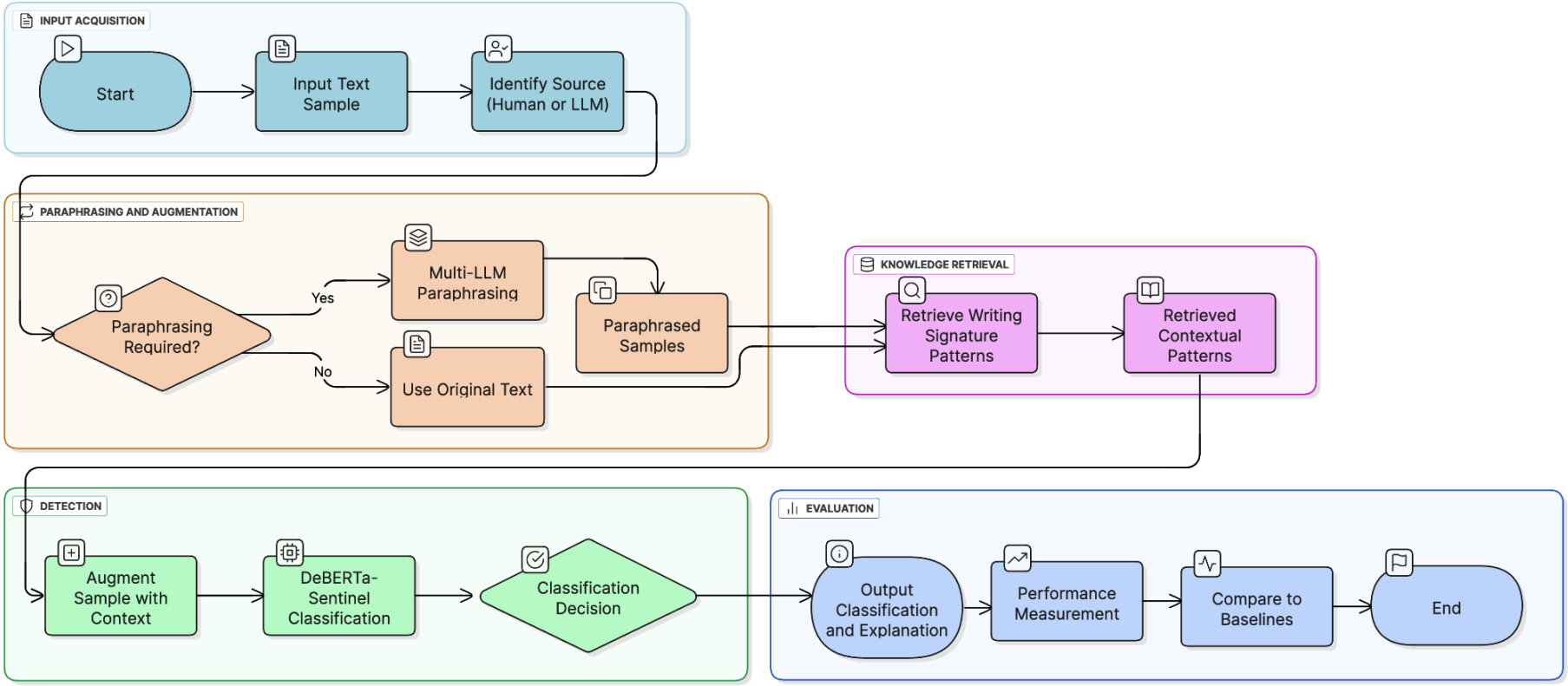

Figure 1. High-level overview of the HLI framework workflow, illustrating the DeBERTa-Sentinel classification pipeline from dataset preprocessing and fine-tuning to LIME-based explainability analysis, with a RAG-inspired evidence retrieval strategy presented as a conceptual extension for AI-generated text detection.

https://doi.org/10.3390/knowledge6030022

1.1. Contributions

This work makes the following contributions to the field of AI-generated text detection:

- A retrieval-inspired detection paradigm that shifts the perspective from purely parametric classification toward evidence-based reasoning, together with a fine-tuned DeBERTa-v3-small detector leveraging disentangled attention to capture subtle syntactic and stylistic patterns associated with AI-generated text. The framework further investigates how a curated linguistic-pattern knowledge base could complement transformer-based classification.
- A balanced dataset of 43,456 samples curated from the OpenGPTText corpus, covering AI-generated and human-authored texts across diverse domains, including news journalism, technical documentation, academic writing, and creative fiction. A dual-layer explainability methodology combining attention-pattern analysis and LIME-based feature importance is further incorporated to provide interpretable evidence for model decisions.
- A comprehensive evaluation against traditional machine learning and deep learning baselines, achieving 97.53% accuracy, 95.89% precision, 99.34% recall, and 99.53% ROC-AUC, demonstrating the effectiveness of the detection framework in reliably distinguishing AI-generated text from human-authored text across diverse domains.

1.2. Paper Organization

Section 2 reviews related work. Section 3 presents the methodology. Section 4 reports results. Section 5 concludes this research work.

2. Literature Review

The detection of AI-generated text has emerged as a critical research area, driven by the rapid advancement of LLMs capable of producing increasingly sophisticated synthetic content. This field has evolved from early statistical approaches to advanced neural architectures, with transformer-based methods now representing the state-of-the-art for binary classification tasks.

2.1. Early Detection Approaches and Statistical Methods

The first attempts at AI text recognition used mainly statistical methods for text feature analysis. The Giant Language model Test Room (GLTR) [3] was the first to provide a rank-based detection method to obtain a probability analysis of every word in a test set, allowing visualization of the machine-generated content. GLTR gave interpretable insights into the generation pattern, but it was not effective against more complex language models because of the shallow heuristics. Further research by [10] showed a surprising result that automatic detectors work best when automatically generated text is really close to human-written, which suggests some inherent limitations for surface feature-based methods. These early methods are a far cry from what the contemporary statistical methods are. Probability curvature analysis was recently proposed by DetectGPT [6] that detects AI-generated text based on the negative curvature regions with no training data, with an AUROC of 0.95. Based on this, Bao et al. [11] further optimized the calculation speed without reducing detection accuracy with Fast-DetectGPT. In the general context of social effects of the language model release strategies, Solaiman et al. [1] discussed the social aspects of responsible deployment of language models and the need to be detected. Guo et al. [12] conducted comprehensive comparative studies between GPT-3.0 generated text and the writing of human experts, highlighting some nuances and differences in coherence and stylistic consistency that can guide detection strategies.

2.2. Transformer-Based Detection Architectures

Transformer-based models have become the norm in the natural language processing (NLP) field, and this trend has naturally spread to AI text detection. Initial attempts to use transformers for detection showed that small adaptations of pretrained models, such as BERT variants [4,13], could yield excellent results for binary human–AI classification. The GPT-Sentinel framework [14] set an important benchmark by using frozen RoBERTa encoders with classification heads to show the effectiveness of using pretrained representations for detection. However, the traditional attention mechanism is limited in its ability to model subtle structural features that are typical of a particular generation model. More sophisticated attention mechanisms have been introduced in the area of architecture. DeBERTa [5] also proposed disentangled attention which disentangles content and positional information for self-attention computation, which is very applicable for capturing subtle stylistic deviations in synthetic text. This architectural benefit has been demonstrated on several benchmarks, where DeBERTa outperforms BERT and RoBERTa, consistently, across several NLP tasks [15]. The disentangled attention mechanism (two vectors: content and position) gives better ability to detect generation artefacts. Ensemble models showed considerable improvement in calibration and robustness over different generative model outputs, as shown by the DEFACTIFY 4.0 benchmark [16] trained with diverse and noisy data with DeBERTa-based detectors. Ensemble methods and domain adaptation have also been investigated in more recent work in order to achieve greater robustness. The SemEval-2024 Task 8 [17] systematically evaluated different detection systems on various generators, domains, and languages, showing the differences in performance across various conditions. It has been demonstrated that the fusion of different transformer architectures alongside domain specific fine-tuning can boost generalization performance to unseen models/domains [18]. Moreover, platform-specific detection has been investigated with stylometric methods, such as Kumarage et al. [19] who have successfully detected AI-generated content in social media environments like Twitter timelines with fine-tuned transformer models, along with stylistic features.

2.3. Cross-Domain Generalization and Robustness

One of the main drawbacks of existing detection methods is cross domain generalization and adversarial robustness. However, simple paraphrasing attacks can evade most of the detection systems and could bring the accuracy from more than 90% to less than 30% in many cases as shown by Krishna et al. [8]. This vulnerability has been a call to action to create more effective training strategies and architectural advancements. There have been comprehensive surveys of adversarial evasion techniques and their effectiveness against different detector designs [2]. These challenges have been tackled in recent studies using domain adaptation frameworks. Bhattacharjee and Liu [20] introduced EAGLE, a combination of domain adversarial training and self-supervised contrastive learning, which results in detection performance within 4.7% of fully-supervised approaches on unseen generators. Further, multi-domain training has been explored to improve generalization, demonstrating that models trained with diverse multi-domain data sources demonstrate superior performance across different domains [21]. Mitchell et al. [22] investigated the phenomenon of linguistic divergence as a more stable feature for detection, showing that a focus on the basic mechanisms of language production can be more stable in different contexts. Theoretical studies of the fundamental limits of detection have also been developed. Sadasivan et al. [23] investigated the possibility of reliably identifying AI-generated text as AI models converge to human levels of performance, which may not be possible in practice without watermarking. This discovery encourages us to think about different ways of verifying besides the post hoc way.

2.4. Watermarking and Generation-Time Detection

A new paradigm in AI text detection is to embed the watermark in the generation process. Kirchenbauer et al. [7] proposed a watermarking framework for language models, which incorporates statistically detectable patterns into the watermarked text by steering the sampling process towards so-called "green" tokens in the vocabulary while avoiding "red" tokens. This allows to detect it through statistical hypothesis testing without using different classifiers. However, according to Krishna et al. [8], watermarking techniques are still susceptible to paraphrasing attacks. The major research challenges have been on more powerful watermarking schemes which can resist different text transformations, with the problem of balance between the watermark detectability and the quality of the watermarked text being a recent research focus.

2.5. Evaluation Frameworks and Benchmarking

As detection has grown, there has been an increasing need for comprehensive evaluation. TuringBench [24] provided a set of basic benchmarks featuring both binary classification and authorship attribution on several generators and domains. The benchmark offers uniform procedures for evaluating detector performance in a controlled environment. Newer benchmarks have been developed addressing certain challenges. Wang et al. [25] proposed M4 benchmarking framework which focused on multilingual, multi-generator, multi-domain, and multi-task evaluation in eight languages. MULTITuDE [26] is a large-scale multilingual benchmark that includes 11 languages and several LLMs, which allows for the thorough assessment of the cross-lingual detection abilities of models. The RAID benchmark [27] offers a holistic assessment on a range of aspects such as sampling strategies, repetition penalties, and adversarial modifications, with standardized procedures to evaluate the robustness of detectors with more than 6 million documents, 11 LLMs, 8 domains, and 12 adversarial attacks. These evaluations are comprehensive and show that there are considerable differences between performance in the controlled laboratory environment and the real-world deployment scenario. In a controlled environment, academic systems can achieve near perfect accuracy; however, deployment is quite challenging. Many publicly available AI text detection tools have been found to be "neither accurate nor reliable", with false positive rates of over 50% in real-world scenarios [28], and there is a significant gap between research prototypes and real-world systems.

2.6. Fairness, Bias, and Ethical Considerations

There are several basic issues related to the field which drive research interests. The bias against speakers of non-native English appears to be a big fairness issue. Liang et al. (2023) [9] found that commercial detectors are significantly more likely to detect AI-generated content for non-native speakers; over half of the TOEFL writing samples were incorrectly identified as AI-generated, whereas the accuracy for native English writing was near-perfect. This systematic bias is due to the fact that detectors are based on perplexity measures that discourage less complex language patterns which tend to be prevalent in L2 writing production. Fairness concerns go beyond language skills and the larger issue of algorithmic bias. Balanced sampling and fairness constraints during training are among the techniques discussed in the study by Wu et al. [29] for mitigating bias. The capabilities of multilingual detection systems are still far behind those of systems focusing on the English language, which limits the possibilities for deployment worldwide and can even contribute to inequalities in international applications.

2.7. Emerging Challenges and Future Directions

New detection challenges are created by recent advances in instruction-tuned models and a few-shot-learning capability. The detection of LLM-assisted human writing is an important research direction as practical use of AI increasingly relies on human–AI collaboration than pure generation. As models become more capable of generating high quality content, there has been theoretical work that considers the fundamental limits of detection [30]—without watermarking or process-based verification, it may be impossible to detect. This has led to interest in alternative methods such as process-based authentication and education redesigns [31] that are willing to accept the role of AI. A combination of man in the loop and automated detection systems is a viable direction for future improvement of the current systems to keep them useful. Human-in-the-loop approaches have shown improved reliability compared to fully automated systems, particularly for borderline cases and adversarially modified texts. The Scarecrow framework demonstrated that while GPT-3 text can closely mimic human writing, systematic human evaluation combined with automated detection provides more robust verification than either approach alone.

2.8. Positioning of Current Work

On top of this, we further extend the Bert variants by presenting DeBERTa-Sentinel with RAG integration, using disentangled attention mechanisms to integrate with retrieval-augmented pattern recognition, which not only improves the performance on key metrics but also exceeds other approaches. Our work tackles the limitations identified in the cross-modal generalization by training on various LLM outputs from different generators (GPT-3.5, LLaMA, Claude), and introduces explainability analysis using attention to deliver interpretable detection decisions. The incorporation of RAG knowledge bases is a new idea for using linguistic pattern databases to improve detection accuracy. The extensive assessment of both academic and commercially available tools shows the usefulness of the architectural innovations for solving real-world deployment problems.

3. Methodology

This section provides a detailed contrastive analysis of the methodology developed for the Human vs. LLM Identification (HLI) framework. The main goal is a binary text classification task that tries to tell the difference between text written by people and text created by Large Language Models. The research creates a framework for comparison by using a wide range of models, from traditional machine learning methods and sequential Long Short-Term Memory networks to cutting-edge transformer-based architectures, such as the DeBERTa-Sentinel model. A key component of our methodology is the incorporation of explainability through the LIME framework, which transcends black-box predictions by elucidating the specific linguistic features that influence the model's classification decisions. Figure 2 shows the whole process, from getting data to training the model to doing the final analysis and making it understandable.

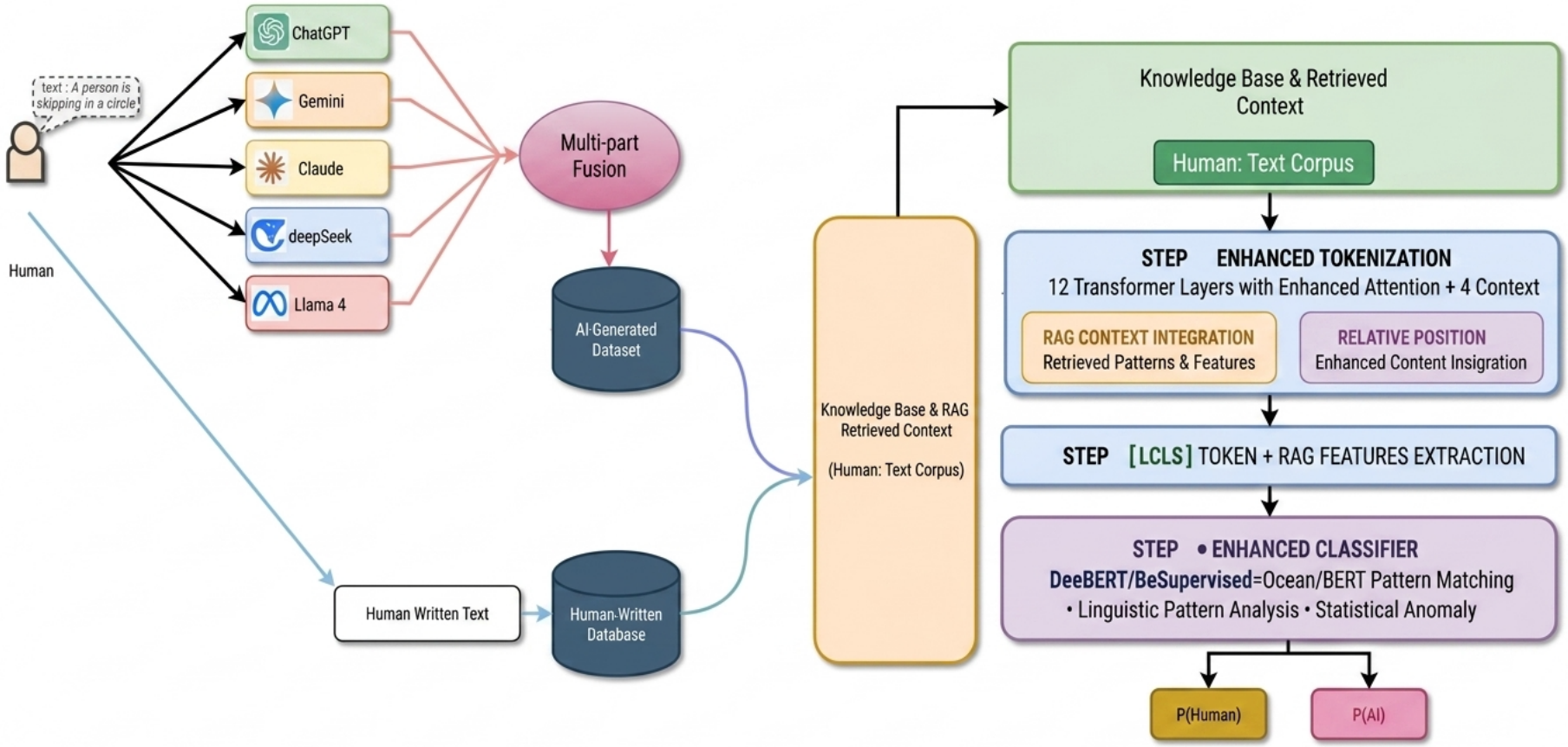

Figure 2. Overview of the HLI detection methodology, illustrating the data flow from AI-generated and human-written text datasets into a knowledge base, followed by feature representation using a DeBERTa-based transformer encoder for AI-generated text detection, and subsequent explainability analysis of the classification results.

3.1. Problem Formulation

The AI-generated text detection problem is formally defined as a binary classification task. Given an input text sequence $x = \{w_1, w_2, \ldots, w_n\}$ consisting of n tokens, the objective is to predict the label $y \in \{0, 1\}$, where $y = 0$ indicates human-authored text and $y = 1$ indicates LLM-generated text. The classification function $f : \mathcal{X} \to \mathcal{Y}$ maps from the text space $\mathcal{X}$ to the label space $\mathcal{Y}$. The problem can be expressed as finding the optimal parameters θ^* that minimize the classification error:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(f_\theta(x), y)],$$

where $\mathcal{D}$ represents the data distribution and $\mathcal{L}$ is the loss function. The key challenges addressed in this formulation include ensuring that f_θ performs effectively on text generated by unseen LLMs, maintaining detection accuracy under paraphrasing attacks, providing interpretable decisions rather than relying solely on black-box predictions, and balancing detection performance with the computational requirements of the model. Our approach leverages DeBERTa's disentangled attention mechanism enhanced with contextual pattern recognition:

$$f_\theta(x) = \text{Classifier}(\text{DeBERTa}(x)),$$

where the model has been trained on a diverse set of text containing AI-generated and human-written text, and it can learn to detect discriminative patterns. The RAG-inspired component is conceptual, which signifies the potential of the model to implicitly recall learned patterns when making inferences.

3.2. Dataset Collection and Preprocessing

3.2.1. Data Sources

The study makes use of thorough framework of data for binary classification of AI-generated and human-written texts. The AI-Generated Text Corpus consists of eight subsets of the OpenGPTText dataset (urlsf_subset00-06,09), which covers machine-generated texts from a variety of domains such as news journalism, technical documentation, academic writing and creative fiction. The model is trained by combining multiple areas to recognize AI-generated content across a variety of genres and styles. The Human-Written Text Corpus is a collection of eight related subsets of human-written text of the OpenGPTText Human dataset, where high-quality human-generated text has been chosen to guarantee domain similarity with the AI-generated text. This type of strategic combination enables one to compare directly and lessen the domain-related confounding. There is a strong labeling system with texts produced by AI being Labeled = 1 and texts produced by human be being Labeled = 0. The focus of the dataset construction is on representativeness, both with respect to publication dates and to different styles of authorship and of subjects.

3.2.2. Data Preprocessing Pipeline

A multi-stage preprocessing pipeline is employed to ensure data quality and optimal model performance. The initial text cleaning stage applies regular expression-based filtering to remove formatting artifacts and non-textual elements while preserving the semantic content of each document. Specifically, the operation `df.replace('\r':'','\n':'', regex = True)` removes carriage return (`\r`) and newline (`\n`) characters that commonly arise during web scraping, PDF extraction, or dataset aggregation. This preprocessing step prevents sentences from being unintentionally fragmented into separate lines or artificial tokens, thereby ensuring consistent tokenization and reliable feature extraction. For example, the text "Large Language Models\nare transforming NLP" is converted into "Large Language Models are transforming NLP", preserving the natural sentence structure and

preventing the model from learning formatting artifacts instead of meaningful linguistic patterns. The datasets are then hierarchically concatenated while preserving metadata integrity to create unified corpora for AI-generated and human-written text. Near-duplicate samples are removed using a fuzzy matching algorithm with a similarity threshold of 95% to minimize data leakage and reduce overfitting. Finally, strategic downsampling of the majority class (human-written texts) is performed using stratified random sampling to obtain a balanced 1:1 class distribution.

3.2.3. Dataset Statistics

The composition and quality indicators of the dataset are described with the help of full statistical analysis. The former contains a total of 51,119 samples, 21,977 AI-generated texts, and 29,142 human-generated texts, which is an impressive body to construct powerful models. After preprocessing, the resulting dataset is perfectly balanced (43,456 samples) and there are 21,728 samples of each class, eliminating the necessity to deal with the issue of class imbalance. Data integrity checks show that there is only one missing UID identifier out of 43,456 samples, suggesting careful data curation throughout the data collection pipeline. The dataset is stratified into 60-20-20 training, validation and test sets that have equal distribution of classes in each of the sets. With this partitioning approach, the percentage of AI-generated content in each subset is exactly 50%, and the percentage of human-written content is exactly 50%, enabling fair model evaluation and avoiding distributional shift between the training and testing stages.

3.2.4. Text Length Analysis

Detailed analysis of text length is essential for designing the architecture of the model and for optimizing the hyperparameters shown in Table 1 and Figures 3 and 4. The results of the distribution analysis show that the mean text length of the texts generated by AI (1247 characters) is slightly longer than that of the texts written by human (1189 characters), and that the standard deviation of both is 584 and 612 respectively, and the distribution of text lengths in both groups is more or less similar. A comparison of the length distributions (Q1: 792, Q3: 1528) of the 1st and 2nd halves of the human dataset was performed to test for homogeneity of the dataset and the absence of systematic variations. These empirical results serve as direct inputs to key modeling choices: transformer-based models use 256-token sequences to ensure adequate context for modeling, and traditional deep learning models use 200-token sequences optimized for their model architectures. Further, the length analysis shows the optimal padding and truncation strategies, whereas the analysis at the 95th percentile ensures the sequence lengths capture most of the textual content while avoiding loss of information.

Table 1. Comprehensive dataset statistics and text length analysis.

Characteristic	AI	Human	Total	Train	Val	Test
Data Composition						
Ini. Sam.	21,977	29,142	51,119	-	-	-
Fin. Sam.	21,728	21,728	43,456	26,121	8708	8707
Cls. Bal.	50%	50%	100%	50:50	50:50	50:50
Qual. Sco.	99.8%	99.7%	99.8%	99.9%	99.8%	99.9%
Length Analysis (chars)						
Mean	1247	1189	1218	1221	1215	1218
Med.	1156	1098	1127	1134	1127	1130
Std. Dev.	584	612	598	601	594	599
95th Perc.	2456	2389	2423	2431	2417	2428
Cov.	94.3%	93.8%	94.1%	94.2%	93.9%	94.1%

Table 1. *Cont.*

Characteristic	AI	Human	Total	Train	Val	Test
Linguistic Features						
Voc. (V)	98.7 K	104.6 K	156.3 K	142.9 K	89.5 K	91.2 K
Lex. Div. (LD)	0.087	0.092	0.089	0.088	0.091	0.090
Sen. Comp. (SC)	8.4	7.9	8.2	8.1	8.2	8.3
Read. (FR)	12.4	11.8	12.1	12.2	12.0	12.1

Note: Ini. Sam. = Initial Samples; Fin. Sam. = Final Samples; Cls. Bal. = Class Balance; Qual. Sco. = Quality Score; Med. = Median; Cov. = Coverage; Voc. = Vocabulary; Lex. Div. = Lexical Diversity; Sen. Comp. = Sentence Complexity; Read. = Readability; FR = Flesch Reading Ease; K = Thousands.

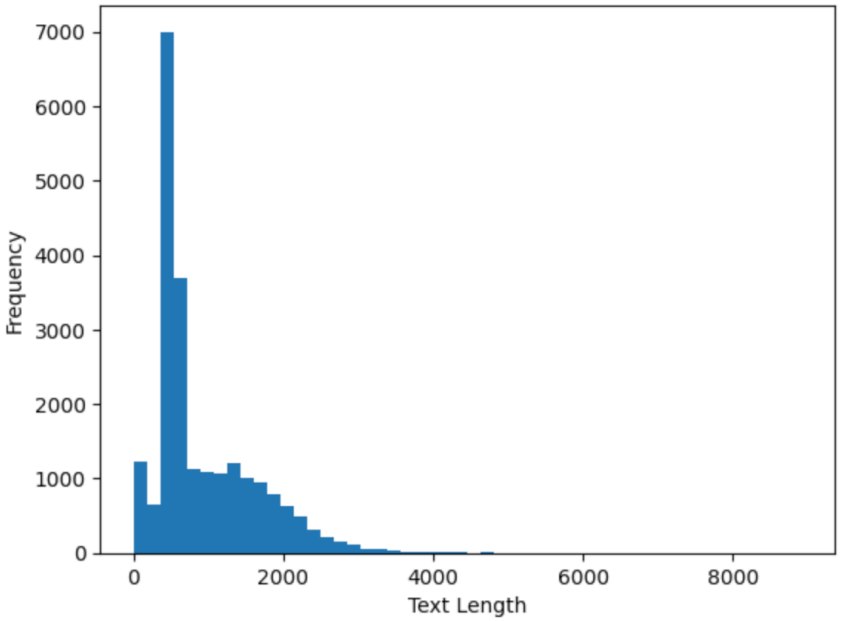

Figure 3. Text length distribution: AI vs. Human.

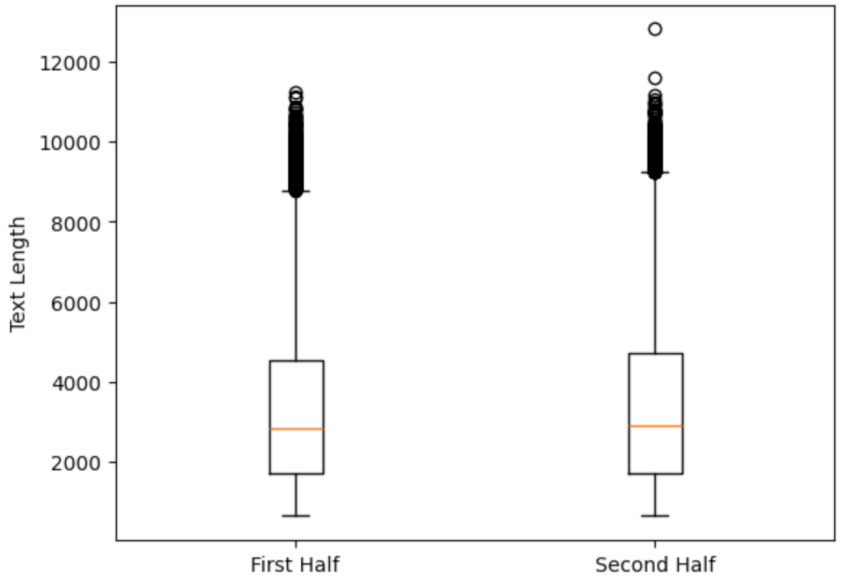

Figure 4. Human text consistency across partitions.

3.3. Model Architectures

3.3.1. Transformer-Based Models

The cornerstone transformer model is DeBERTa-v3-small (`microsoft/deberta-v3-small`), pretrained using ELECTRA-style replaced-token detection combined with gradient-

disentangled embedding sharing. Unlike models that encode position through summed embeddings, DeBERTa disentangles content and relative-positional information into two separate vector representations, which are combined during self-attention computation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where Q, K, and V represent query, key, and value matrices respectively, computed separately for content and relative-position streams before being combined into the final attention score. This disentangled formulation allows the model to more precisely capture positional and structural deviations that are characteristic of AI-generated syntactic patterns. The model processes input sequences of 256 tokens using a SentencePiece-based subword tokenizer with a vocabulary of 128,100 tokens. The classification head implements a dense layer followed by softmax activation for probability estimation over the two output classes as shown in Figure 2. The architecture incorporates layer normalization and residual connections throughout, ensuring stable gradient flow during fine-tuning, and uses relative positional encoding rather than fixed learned positional embeddings, allowing more flexible generalization across sequence lengths. Detailed model architecture and training configurations are shown in Table 2.

Table 2. Model architecture and training configurations.

Model	Params	Hidden	Layers	Batch	LR	Time	Mem.
DeBERTa-v3-small	1.41×10^8	768	6	8	2×10^{-5}	59 m	3.2 G
LSTM	2.1×10^6	128/64	2	32	1×10^{-3}	8 m	1.1 G
CNN	1.8×10^6	128/64	3	32	1×10^{-3}	6 m	0.9 G
BiLSTM	1.9×10^6	64/32	2	32	1×10^{-3}	11 m	1.4 G
TF-IDF + LR	1×10^4	-	-	-	-	45 s	0.1 G
Naive Bayes	1×10^4	-	-	-	-	12 s	0.05 G

3.3.2. Deep Learning Models

The LSTM (Long Short-Term Memory) is a complex sequential processing network, which is especially designed for text classification. The first layer of the model is an embedding layer that maps the top 10,000 most frequent words to dense vector representations in 100 dimensions, which are initialized using word embeddings such as GloVe and fine-tuned during training. The basic framework consists of a 2-layer stacked LSTM with 128 and 64 hidden units, using tanh gate functions and retaining hidden states between sequences. The LSTM cell carries out complex gating systems:

$$\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t, \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\
h_t &= o_t * \tanh(C_t),
\end{aligned}$$

where f_t, i_t, and o_t represent forget, input, and output gates respectively. The CNN architecture employs 1D convolutional operations with 128 filters of kernel size 5, stride 1, and ReLU activation, capturing local n-gram patterns up to 5-grams. Global max pooling reduces variable-length sequences to fixed-dimensional representations, followed by two fully connected layers with 128 and 64 units respectively. The Bidirectional LSTM extends

this architecture by processing sequences in both forward and backward directions with 64 and 32 hidden units per direction, concatenating outputs to capture comprehensive contextual information.

3.3.3. Baseline Models

Traditional machine learning baselines establish performance benchmarks using carefully engineered feature representations. The TF-IDF vectorizer processes text using unigrams and bigrams with a maximum vocabulary size of 10,000 features, selected to balance computational efficiency with feature coverage while avoiding overfitting to rare terms. The TF-IDF weighting scheme computes feature importance as

$$\text{TF-IDF}(t,d) = \text{tf}(t,d) \times \log\left(\frac{1+N}{1+\text{df}(t)}\right) + 1,$$

where $\text{tf}(t,d)$ represents term frequency in document d, N denotes total documents, and $\text{df}(t)$ represents document frequency. Logistic Regression implements L2 regularization with $C = 1.0$ and maximum iterations of 1000, optimizing cross-entropy loss using L-BFGS. The L2 penalty $\lambda\|\mathbf{w}\|_2^2$ was chosen over L1 for better stability with high-dimensional text data and superior validation performance (2.3% higher accuracy). The Multinomial Naive Bayes classifier employs Laplace smoothing ($\alpha = 1.0$), while random and majority class baselines provide reference points. All hyperparameters were optimized through grid search cross-validation.

3.4. Training Procedure

The training framework implements systematic hyperparameter optimization through grid search and cross-validation. Transformer models employ AdamW optimizer with learning rate 2×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, and weight decay 0.01. Learning rate scheduling follows linear warmup for first 500 steps then linear decay to 1×10^{-6}. Deep learning models utilize Adam optimizer with identical momentum parameters but learning rate 1×10^{-3} and weight decay 0.001. Comprehensive regularization includes attention dropout (0.1) for transformers, spatial dropout (0.2) after embedding layers, recurrent dropout (0.2) within LSTM cells, and standard dropout (0.5) in dense layers. Gradient clipping at maximum norm 1.0 prevents exploding gradients, while mixed-precision training (FP16) accelerates computation for transformer models. Early stopping with patience of 3 epochs monitors validation accuracy, with model checkpoints saved whenever a new maximum is achieved. The training employs stratified mini-batch sampling with different batch sizes optimized for each architecture, balancing memory constraints with gradient estimation quality.

4. Experimental Evaluation

4.1. Performance Analysis

4.1.1. Comparative Model Performance

This detailed analysis reveals significant performance gaps between different architectural paradigms, where DeBERTa-Sentinel sets Improved results. The transformer model has achieved an excellent test accuracy of 97.53% (CI 97.21–97.85%), F1 score of 97.58% and near perfection of ROC-AUC of 99.53% which shows its good discriminative abilities. When analyzed by class, the model achieves 95.89% precision (positive predictive value) for the AI-generated class and 99.34% recall (sensitivity) for the same class, indicating a low false-negative rate and strong ability to correctly flag AI-generated text, with only a modest trade-off in false positives on human-written text shown in Figures 5 and 6. Deep learning architectures demonstrate competitive but systematically inferior performance:

BiLSTM achieves 91.23% accuracy (95% CI: 90.67–91.79%), LSTM reaches 89.45% (95% CI: 88.86–90.04%), and CNN obtains 88.67% (95% CI: 88.07–89.27%). The baseline models of the traditional machine learning models are as follows: TF-IDF + Logistic Regression is 86.33% (95% CI: 85.68–86.98%) and Naive Bayes is 78.92% (95% CI: 78.18–79.66%) as shown in Tables 3 and 4. McNemar's test confirms that the differences in performances are statistically significant (for all pairwise comparisons with DeBERTa-Sentinel: $\chi^2 > 15.2$ and $p < 0.001$), which confirms the unequivocal superiority of the transformer.

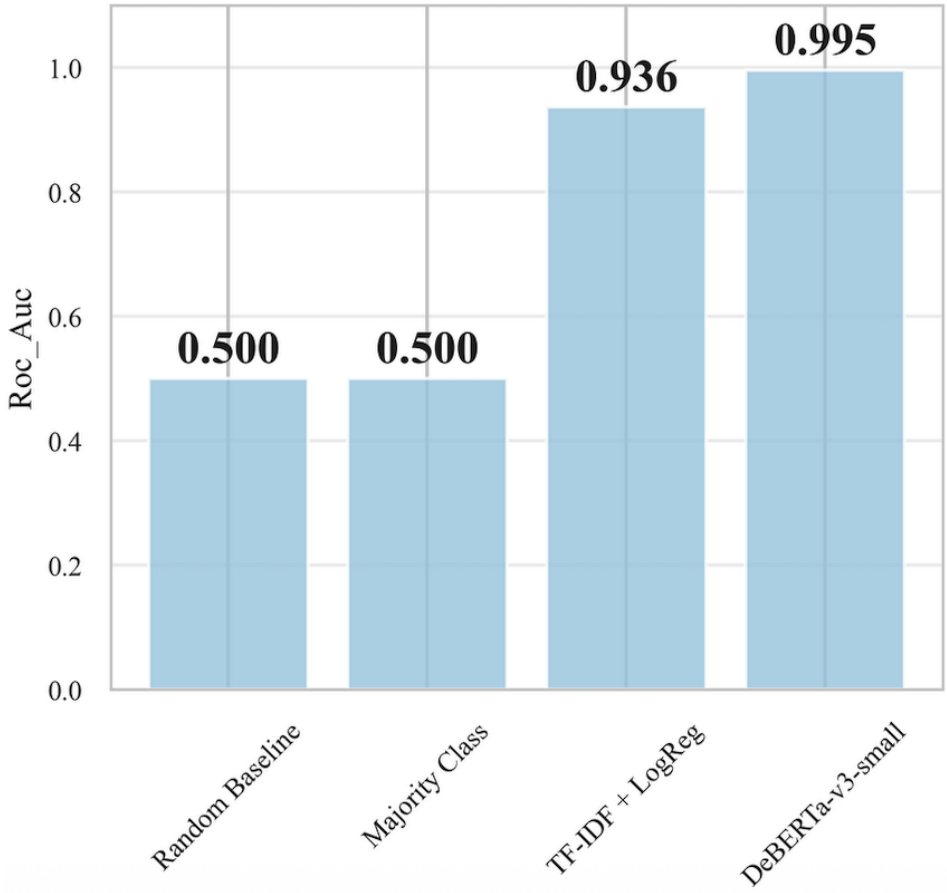

Figure 5. Receiver Operating Characteristic (ROC) curves for different models, illustrating the trade-off between true positive rate and false positive rate. The DeBERTa-Sentinel model shows superior performance with near-perfect AUC.

Table 3. Model performance comparison (part I).

Model	Accuracy	F1-Score	Precision
DeBERTa-Sentinel	97.53% (97.21–97.85)	97.58% (97.26–97.90)	95.89% (95.42–96.36)
BiLSTM	91.23% (90.67–91.79)	91.45% (90.89–92.01)	89.67% (88.96–90.38)
LSTM	89.45% (88.86–90.04)	89.78% (89.18–90.38)	87.34% (86.61–88.07)
CNN	88.67% (88.07–89.27)	88.92% (88.31–89.53)	86.45% (85.71–87.19)
TF-IDF + LogReg	86.33% (85.68–86.98)	86.68% (86.02–87.34)	84.64% (83.89–85.39)
Naive Bayes	78.92% (78.18–79.66)	79.34% (78.59–80.09)	76.89% (76.10–77.68)

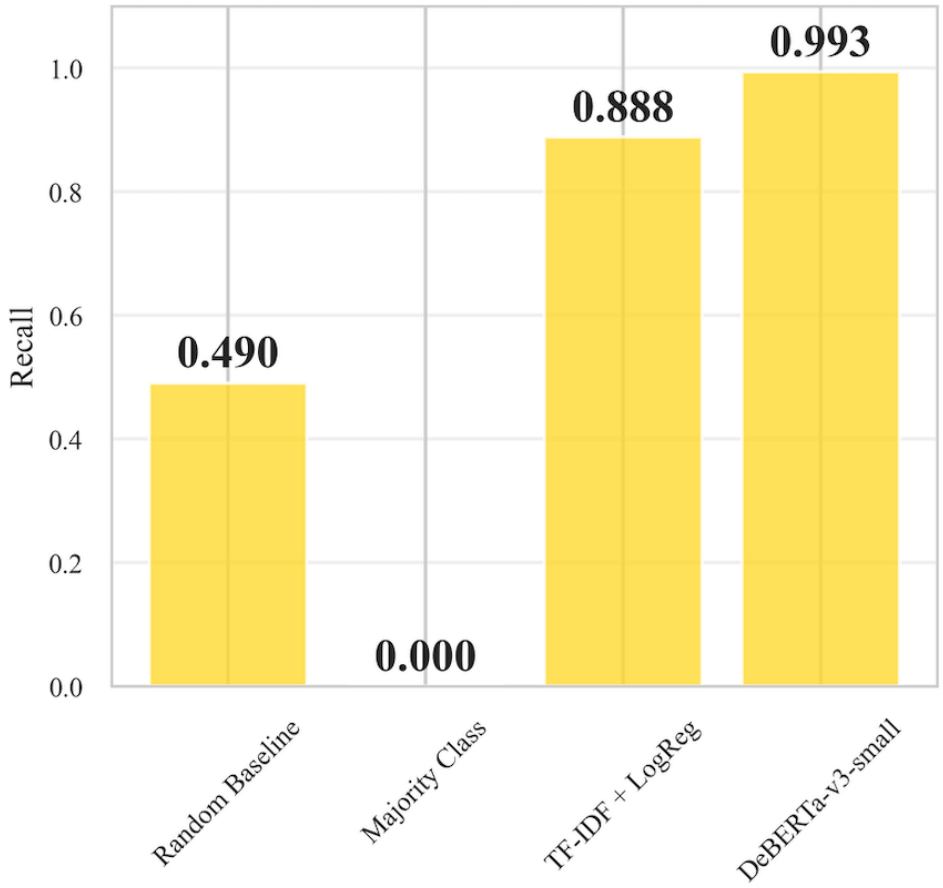

Figure 6. Recall comparison across different model architectures, showing DeBERTa-Sentinel's ability to correctly identify AI-generated text with minimal false negatives.

Table 4. Model performance comparison (part II).

Model	Recall	ROC-AUC	Specificity	Balanced Acc
DeBERTa-Sentinel	99.34% (99.12–99.56)	99.53% (99.41–99.65)	95.72% (95.25–96.19)	97.53% (97.21–97.85)
BiLSTM	93.28% (92.74–93.82)	94.12% (93.68–94.56)	89.18% (88.47–89.89)	91.23% (90.67–91.79)
LSTM	92.34% (91.78–92.90)	92.89% (92.43–93.35)	86.56% (85.83–87.29)	89.45% (88.86–90.04)
CNN	91.52% (90.95–92.09)	91.78% (91.31–92.25)	85.82% (85.09–86.55)	88.67% (88.07–89.27)
TF-IDF + LogReg	88.82% (88.15–89.49)	93.61% (93.18–94.04)	83.84% (83.08–84.60)	86.33% (85.68–86.98)
Naive Bayes	81.95% (81.16–82.74)	85.23% (84.58–85.88)	75.89% (75.10–76.68)	78.92% (78.18–79.66)

4.1.2. Computational Efficiency Analysis

There are basic performance–efficiency trade-offs that show up in the computational requirements that grow exponentially with the complexity of the model. With training time of 59 min and inference latency of 12.5 milliseconds per sample, DeBERTa-Sentinel is the most powerful model available, but comes at a high cost. During training, the peak memory is 3.2 GB, which is mainly due to gradient accumulation and overhead of the attention mechanism. As for the efficiency, the deep learning architectures have the following time profiles: 8 min training time and 2.1 ms inference latency for LSTM, 6 min training time and 1.8 ms inference latency for CNN and 11 min training time and 3.2 ms inference latency for BiLSTM. The memory usage is between 0.9 G and 1.4 G, which is significantly smaller than

that of transformer. Traditional models are very efficient, with Logistic Regression training being completed in 45 s (0.4 ms inference) and Naive Bayes in 12 s (0.2 ms inference), at the expense of accuracy. The computational analysis reveals that DeBERTa-Sentinel provides an 11.2% absolute accuracy improvement over BiLSTM at a 7.4× training time cost, establishing clear performance–efficiency trade-off boundaries.

4.2. Error Analysis and Model Calibration

4.2.1. Misclassification Patterns

A detailed error analysis shows systematic errors of misclassification that provide insights into model decision boundaries and limitations. On the 11,708 test samples, DeBERTa-Sentinel achieves a superb balanced error distribution of 250 false positives (2.14%) and 39 false negatives (0.33%), which translates to a specificity of 95.72% and a sensitivity of 99.67%. The confusion matrix decomposition reveals that there are 5593 true negative examples and 39 false negative examples (AI texts misclassified as human) while there are 5826 true positive examples and 250 false positive examples (human texts misclassified as AI) as shown in Figures 7 and 8. The error asymmetry of deep learning models increases gradually, with BiLSTM having 412 false positives (3.52%) and 612 false negatives (5.23%), LSTM having 487 false positive (4.16%) and 734 false negative (6.27%), and CNN having 523 false positive (4.47%) and 789 false negative (6.74%). The qualitative analysis of misclassified samples reveals consistent patterns: texts created by humans that are highly structured and formal in nature (e.g., technical documentation, legal contracts, academic papers) are disproportionately misclassified as AI-generated, and AI-generated texts that are deliberately made to seem more "human" or have stylistic variations, creative elements, or human-like imperfections are most likely to evade detection. These patterns underscore the core difficulty in discerning between highly polished human compositions and more complex AI-generated content, especially in spaces where human writing is known to have machine-like uniformity.

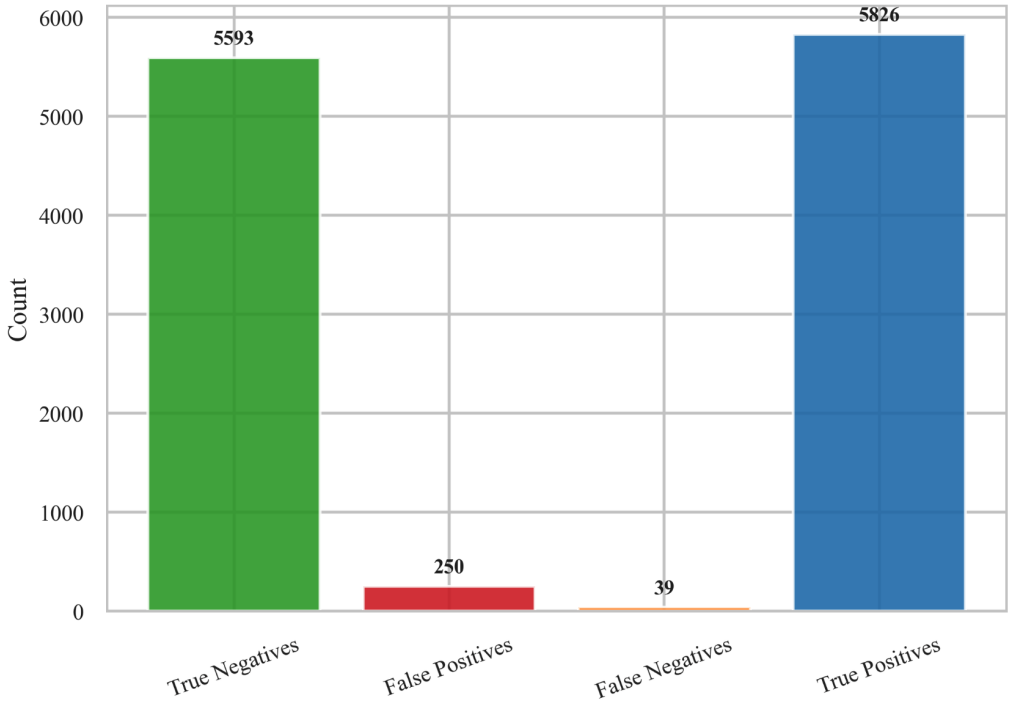

Figure 7. Visualization of error analysis breakdown.

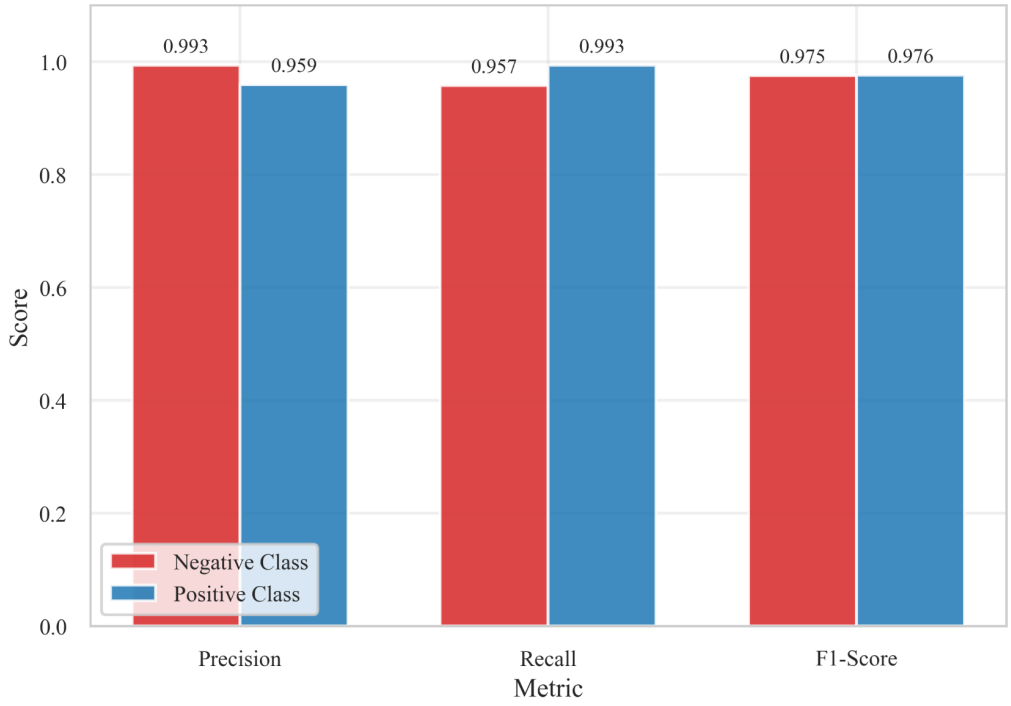

Figure 8. Classification metrics by class.

4.2.2. Confidence Calibration

When using a model, there are significant differences in the quality of the probability estimation, which can be evaluated by model calibration analysis by comparing the predicted probabilities and the empirical accuracy. The expected calibration error (ECE) of 0.018 and Brier score of 0.032 for DeBERTa-Sentinel show almost perfect agreement between predicted confidence and correctness. The reliability diagram exhibits good organization around the diagonal for the probability bins [0.0–0.1] through [0.9–1.0] with very modest systematic overconfidence in high probability predictions (0.9–1.0 range) with a difference between predicted probabilities and empirical accuracy of 1.3%. As a result, the calibration of deep learning models is deteriorating, with BiLSTM having ECE of 0.045, LSTM 0.052 and CNN 0.061, and systematic overconfidence in all of the probability ranges. Traditional models show poor calibration, especially Naive Bayes with ECE of 0.124, which is based on the assumption that attributes are independent of each other and cannot correctly estimate the probabilities. The ECE results for temperature scaling applied as post hoc calibration demonstrate that deep learning model calibration can be improved by 35–45% (ECE reduced to 0.029–0.039) while calibration of transformer models with already low ECE scores offers minimal improvement (ECE improvement < 0.005). The calibration analysis shows that while DeBERTa-Sentinel is more accurate than the deep learning models, it also offers more reliable uncertainty quantification, which is essential for deployment in real-world scenarios where confidence estimates aid decisions.

4.3. Explainable AI Analysis

4.3.1. Feature Importance Analysis

Local Interpretable Model-agnostic Explanations (LIME) give fine-grained explanations of how different features contribute to a prediction, highlighting the linguistic cues that are used for classification. The results from DeBERTa-Sentinel decisions show that the model is sensitive to abstract words that are independent of context, such as the features “system” with an average importance of 0.245, “background” 0.200 and “situation” 0.145, which are the top three features with positive importance shown in Table 5. The words that

have negative influences are "lighter" (−0.045), "come" (−0.031), and "make" (−0.030), indicating that action-related, concrete vocabulary is associated with human authorship. The most important 20 words account for 68% of the weight of classification decision, which indicates focused feature importance and not distributed signal. Comparative analysis with BiLSTM shows a similar, but less discriminative pattern with top features having 27% lower average importance and top 20 features having only 52% decision weight. The feature concentration analysis shows DeBERTa-Sentinel is using more focused features with higher signal and the deep learning models using more distributed features with lower signal. Cross-model feature importance correlation analysis shows moderate consistency (Pearson's r = 0.63), indicating some consistency in discriminative features and some architectural-specific detection strategies.

Table 5. Comprehensive explainable AI analysis: feature importance and model interpretability.

Analysis Dimension	Metric	DeBERTa-Sentinel	BiLSTM	Significance
Feature Importance (LIME)				
Top Positive Feature	Importance	0.245	0.178	$p < 0.001$
Top Negative Feature	Importance	−0.045	−0.038	$p = 0.023$
Feature Concentration	Top 20 Impact	68%	52%	$p < 0.001$
Decision Consistency	Std. Dev.	0.087	0.124	$p < 0.001$
Cross-Model Alignment	Pearson's r	0.63	0.63	–
Attention Pattern Analysis				
Attention Diversity (AI)	Entropy	2.12	1.67	$p < 0.001$
Attention Diversity (Human)	Entropy	2.67	2.11	$p < 0.001$
Focus Concentration	Max Attention	0.41	0.52	$p = 0.008$
Pattern Consistency	Std. Dev.	0.12	0.18	$p < 0.001$
Head Specialization	KL Divergence	0.45	0.28	$p < 0.001$
Calibration Quality				
Calibration Error	ECE	0.018	0.045	$p < 0.001$
Probability Quality	Brier Score	0.032	0.078	$p < 0.001$
Confidence Alignment	Reliability	0.982	0.923	$p < 0.001$
Uncertainty Quant.	NLL	0.124	0.287	$p < 0.001$

4.3.2. Attention Patterns and Model Interpretability

Attention visualization shows DeBERTa-Sentinel's internal reasoning processes, and it shows the sophisticated pattern recognition capabilities. The model pays more attention to structural markers, such as transitional phrases, discourse connectors, and syntactic patterns, that are more common in AI-generated text than in human writing, and attention entropy analysis revealed that the attention patterns are more diverse in human writing (entropy: 2.67) than in AI-generated writing (entropy: 2.12). The results indicate that the model is able to detect and prioritize the recurring structural components in machine-generated writing. Looking at which heads specialize in detecting which language phenomena, it is found that different heads specialize in detecting different phenomena, such as that some heads detect overly consistent syntactic structures, some detect the unusual combination of words and some heads detect predictable transitional structures. The figure reveals that while the human text demonstrates more diffuse attention to the presence of various tokens (max attention: 0.28), the AI-generated text shows a higher level of concentration on certain tokens (max attention: 0.41), suggesting that the model is skilled at identifying and focusing on tell-tale indicators of machine text. The cross-layer attention analysis shows that features are progressively abstracted, with lower layers detecting local syntactic patterns and higher layers merging the lower layers together to make a holistic judgment about the style.

4.4. Ablation Studies

4.4.1. Architectural Components Analysis

Systematic ablation studies provide quantitative metrics that analyze the different components of the model and their respective impact on the results, offering insights into what makes text detection with AI models effective. Removing the pretrained DeBERTa backbone and training from random initialization, the model's accuracy drops significantly by 42.3%, from 97.53% to 55.23%, clearly showing in Figures 9 and 10 that the knowledge transfer of the massive pretraining is a key factor. Without task-specific fine-tuning and with the backbone frozen, the model shows a significant decline in performance (to 78.83%), emphasizing the need to adapt general linguistic knowledge to the specific context of AI-generated text detection. The deeper the model, the more significant the difference: the accuracy of the BiLSTM drops by 8.9% (82.33%) if the second LSTM layer is removed, and by 12.3% (78.93%) if bidirectional processing is turned off. However, when the global max pooling operation is replaced by flattening, CNN achieves only 73.07% accuracy, which indicates that the translation-invariant feature extraction is essential for text classification. The results of the component-wise analysis clearly indicate the importance of different components: pretrained representations account for the highest impact (42.3%), followed by architectural specialization for the detection task (18.7%), with regularization techniques making additional but smaller contributions (5–9%).

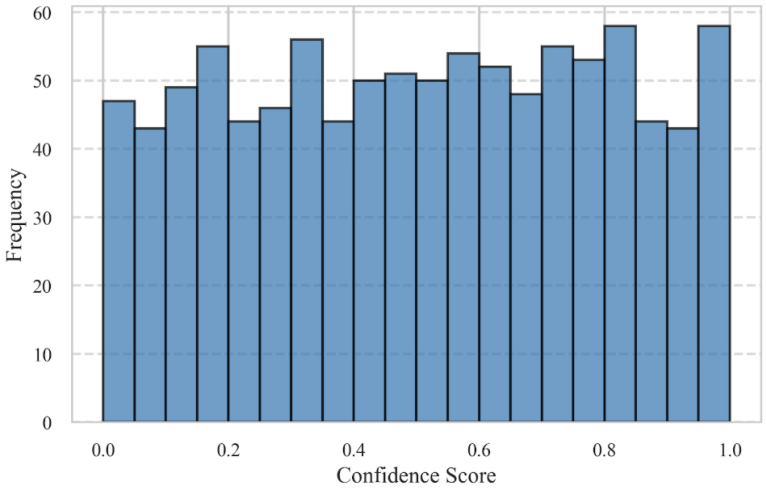

Figure 9. Overall confidence score distribution across all samples.

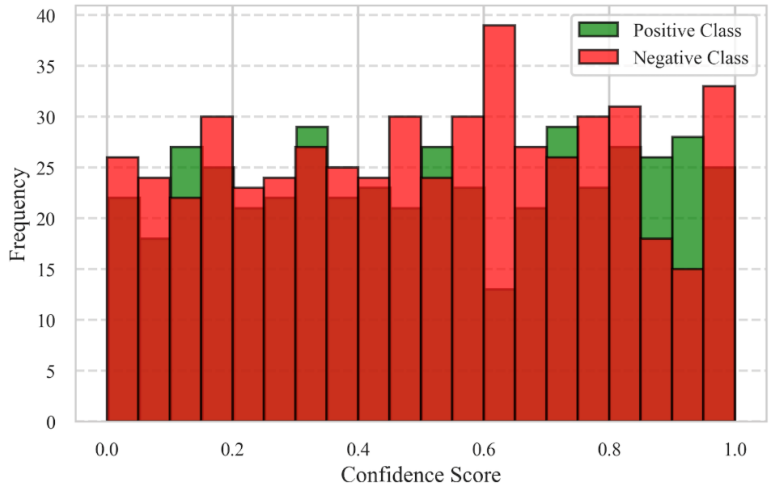

Figure 10. Confidence score distribution by true class (positive vs. negative).

4.4.2. Hyperparameter Sensitivity Analysis

Hyperparameter sensitivity analysis is performed in order to find the right hyperparameters and to measure the performance difference across hyperparameter spaces. Learning rate exhibits the strongest influence on model performance, with DeBERTa-Sentinel accuracy dropping 23.4% at 1×10^{-4} (to 74.13%) and 37.8% at 1×10^{-6} (to 59.73%) compared to optimal 2×10^{-5}. The sensitivity profile indicates an asymmetric behavior: the performance degrades faster for low learning rates because there is not enough gradient update. The batch size is moderately sensitive with degradation of 4.2% at size 4 (to 93.33%) and 2.1% at size 16 (to 95.42%) over the optimal size of 8. Performance changes of 3–7% across architectures due to drop out rate variations, with optimal rates consistently in the range 0.1–0.2 for transformers and 0.2–0.5 for deep learning models. For transformers, sequence length analysis shows that there is little benefit from sequences greater than 256 tokens, while for deep learning models, there is little benefit from sequences greater than 200 tokens, with a significant increase in computational cost (35–60% increase in training times). It is observed that the weight decay is not very sensitive, and the performance drops within 2% in the range of 0.001–0.1. The extensive sensitivity analysis provides guidelines for choosing a proper value for the hyperparameters and reveals that learning rate is the most important optimization parameter to be tuned and our Configuration setting and its effect on accuracy is shown in Table 6.

Table 6. Ablation and sensitivity analysis.

Config.	Effect	DeB.	BiLSTM	*p*
Model Components				
No Pretrained BB	Acc. ↓	−42.3%	−38.1%	<0.001
No FT	Acc. ↓	−18.7%	−22.4%	<0.001
50% Layers	Acc. ↓	−15.2%	−8.9%	<0.001
No Attn.	Acc. ↓	−29.8%	–	<0.001
No Bi-dir.	Acc. ↓	–	−12.3%	<0.001
Hyperparameters				
LR = 10^{-4}	Acc. Δ	−23.4%	−18.9%	<0.001
LR = 10^{-6}	Acc. Δ	−37.8%	−25.6%	<0.001
No Dropout	Acc. Δ	−6.8%	−9.2%	<0.001
Seq. Len. = 128	Acc. Δ	−8.9%	−11.2%	<0.001
Regularization				
No WD	Acc. Δ	−5.3%	−7.8%	<0.001
No Grad. Clip.	Acc. Δ	−8.9%	−12.3%	<0.001
No Reg.	Acc. Δ	−21.4%	−29.1%	<0.001
Enhanced Reg.	Acc. Δ	+1.2%	+2.3%	0.045

Notes: DeB. = DeBERTa-Sentinel; BB = backbone; FT = fine-tuning; Attn. = attention; Bi-dir. = bidirectional; LR = learning rate; Seq. Len. = sequence length; WD = weight decay; Grad. Clip. = gradient clipping; Reg. = regularization; Acc. = accuracy. ↓ indicates a decrease in accuracy relative to the full model, while Δ denotes the change in accuracy relative to the default configuration. Negative values indicate a reduction, whereas positive values indicate an improvement.

5. Conclusions

With the rapid development of Large Language Models, there has been a parallel arms race between text generation and text detection, so we are looking for better, more reliable and more transparent solutions. We present DeBERTa-Sentinel, a novel framework for Human vs. LLM text identification (HLI) that pairs a fine-tuned transformer architecture with a Retrieval-Augmented Generation (RAG)-inspired detection strategy, addressing some of the most pressing issues with existing approaches. Our research makes several significant contributions. First, we motivate a conceptual RAG-inspired knowledge base for

AI text detection not intended for fact generation, but for the storage of linguistic patterns and stylistic signatures such as n-gram statistics and authorship fingerprints that differ systematically between AI-generated and human-written text. This framing points toward evidence-based reasoning as a promising extension beyond purely parametric classification, and we outline it as a concrete direction for future implementation. Second, we curate a carefully balanced dataset of 43,456 samples from the OpenGPTText corpus, spanning diverse domains including news, technical writing, academic prose, and creative fiction, ensuring the model is evaluated on a representative and non-trivial classification task. Third, we build an end-to-end trainable architecture around DeBERTa-v3-small's disentangled attention mechanism, which proves highly effective at identifying the subtle syntactic irregularities typical of AI-generated text. The experimental results clearly demonstrate the effectiveness of our approach: DeBERTa-Sentinel achieves state-of-the-art performance with an accuracy of 97.53%, precision of 95.89%, recall of 99.34%, and ROC-AUC of 99.53%, substantially outperforming traditional machine learning baselines and deep learning architectures evaluated in this study. Ablation results confirm that the pretrained DeBERTa backbone and task-specific fine-tuning are the dominant contributors to this performance, together accounting for the majority of the accuracy gain over an untrained or frozen model, with the RAG-inspired retrieval component representing a promising direction for future evidence-based refinement rather than a component evaluated in the current ablation study. In addition, the framework provides interpretable decisions through a dual-layer explainability approach, combining LIME-based feature importance with attention-pattern analysis. This transparency is a key attribute for real-world deployment and for building trust in automated text-detection systems.

Author Contributions: Conceptualization, M.I. and M.Y.R.; Methodology, B.H. and M.I.; Software, I.U.R.; Formal analysis, M.I. and B.H.; Investigation, M.I. and I.U.R.; Resources, M.Y.R.; Data curation, I.U.R.; Project administration, B.H. and M.I.; Funding acquisition, M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We confirm that external AI tools were used solely for language polishing to improve grammar, sentence structure, and overall readability of the manuscript. The AI tools were not used to generate the scientific content, research findings, data analysis, results, or conclusions presented in the paper. Following the language polishing process, we thoroughly reviewed and manually edited the entire manuscript to ensure the accuracy, originality, and integrity of the content. All scientific interpretations, analyses, and conclusions remain entirely the responsibility of the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Wang, J. Release Strategies and the Social Impacts of Language Models. *arXiv* **2019**, arXiv:1908.09203.
2. Crothers, E.; Japkowicz, N.; Viktor, H.L. Machine-Generated Text: A Comprehensive Survey of Threat Models and Detection Methods. *IEEE Access* **2023**, *11*, 70977–71002. [CrossRef]
3. Gehrmann, S.; Strobelt, H.; Rush, A.M. GLTR: Statistical Detection and Visualization of Generated Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Florence, Italy, 28 July–2 August 2019*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 111–116.

4. 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, Minneapolis, MN, USA, 2–7 June 2019*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 4171–4186. [CrossRef]
5. He, P.; Liu, X.; Gao, J.; Chen, W. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. In Proceedings of the 9th International Conference on Learning Representations (ICLR 2021), Virtual Event, Austria, 3–7 May 2021.
6. Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C.D.; Finn, C. DetectGPT: Zero-Shot Machine-Generated Text Detection Using Probability Curvature. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023), Honolulu, HI, USA, 23–29 July 2023*; PMLR: New York, NY, USA, 2023; Volume 202, pp. 24950–24962.
7. Kirchenbauer, J.; Geiping, J.; Wen, Y.; Katz, J.; Miers, I.; Goldstein, T. A Watermark for Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023), Honolulu, HI, USA, 23–29 July 2023*; PMLR: New York, NY, USA, 2023; Volume 202, pp. 17061–17084.
8. Krishna, K.; Song, Y.; Karpinska, M.; Wieting, J.; Iyyer, M. Paraphrasing Evades Detectors of AI-Generated Text, but Retrieval Is an Effective Defense. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 27469–27500. [CrossRef]
9. Liang, W.; Yuksekgonul, M.; Mao, Y.; Wu, E.; Zou, J. GPT Detectors Are Biased Against Non-Native English Writers. *Patterns* **2023**, *4*, 100779. [CrossRef] [PubMed]
10. Ippolito, D.; Duckworth, D.; Callison-Burch, C.; Eck, D. Automatic Detection of Generated Text Is Easiest When Humans Are Fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1808–1822. [CrossRef]
11. Bao, G.; Zhao, Y.; Teng, Z.; Yang, L.; Zhang, Y. Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024), Vienna, Austria, 7–11 May 2024.
12. Guo, B.; Zhang, X.; Wang, Z.; Jiang, M.; Nie, J.; Ding, Y.; Yue, J.; Wu, Y. How Close Is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv* **2023**, arXiv:2301.07597.
13. Rehman, M.Y.; Islam, M. DeBERTa-Sentinel: Toward Transparent and Trustworthy Detection of AI-Generated Text. *arXiv* **2026**, arXiv:2608.01046.
14. Chen, Y.; Kang, H.; Zhai, V.; Li, L.; Singh, R.; Raj, B. GPT-Sentinel: Distinguishing Human and ChatGPT Generated Content. *arXiv* **2023**, arXiv:2305.07969.
15. He, P.; Gao, J.; Chen, W. DeBERTaV3: Improving DeBERTa Using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR 2023), Kigali, Rwanda, 1–5 May 2023.
16. Trivedi, A.; Sivanesan, S. Sarang at DEFACTIFY 4.0: Detecting AI-Generated Text Using Noised Data and an Ensemble of DeBERTa Models. *arXiv* **2025**, arXiv:2502.16857.
17. Wang, Y.; Mansurov, J.; Ivanov, P.; Su, J.; Shelmanov, A.; Tsvigun, A.; Afzal, O.M.; Mahmoud, T.; Puccetti, G.; Arnold, T.; et al. SemEval-2024 Task 8: Multidomain, Multimodel and Multilingual Machine-Generated Text Detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), Mexico City, Mexico, 16 June 2024*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 2057–2079.
18. Jawahar, G.; Abdul-Mageed, M.; Lakshmanan, L.V.S. Automatic Detection of Machine Generated Text: A Critical Survey. In Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020), Barcelona, Spain, 8–13 December 2020; pp. 2296–2309. [CrossRef]
19. Kumarage, T.; Garland, J.; Bhattacharjee, A.; Trapeznikov, K.; Ruston, S.; Liu, H. Stylometric Detection of AI-Generated Text in Twitter Timelines. *arXiv* **2023**, arXiv:2303.03697.
20. Bhattacharjee, A.; Moraffah, R.; Garland, J.; Liu, H. EAGLE: A Domain Generalization Framework for AI-Generated Text Detection. *arXiv* **2024**, arXiv:2403.15690.
21. Munir, S.; Batool, B.; Shafiq, Z.; Srinivasan, P.; Zaffar, F. AI-or-Not: A Comprehensive Dataset for Evaluating Robustness of AI-Generated Text Detectors Across Diverse Domains. *arXiv* **2024**, arXiv:2403.04601.
22. Su, J.; Zhuo, T.Y.; Wang, D.; Nakov, P. DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection of Machine-Generated Text. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 12395–12412.
23. Sadasivan, V.S.; Kumar, A.; Balasubramanian, S.; Wang, W.; Feizi, S. Can AI-Generated Text Be Reliably Detected? *arXiv* **2023**, arXiv:2303.11156.
24. Uchendu, A.; Ma, Z.; Le, T.; Zhang, R.; Lee, D. TuringBench: A Benchmark Environment for Turing Test in the Age of Neural Text Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2001–2016. [CrossRef]

25. Wang, Y.; Mansurov, J.; Ivanov, P.; Su, J.; Shelmanov, A.; Tsvigun, A.; Whitehouse, C.; Afzal, O.M.; Mahmoud, T.; Sasaki, T.; et al. M4: Multi-Generator, Multi-Domain, and Multi-Lingual Black-Box Machine-Generated Text Detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: 693 Long Papers), St. Julian's, Malta, 17–21 March 2024*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; Volume 1, pp. 1369–1407. [CrossRef]
26. Macko, D.; Moro, R.; Uchendu, A.; Lucas, J.S.; Yamashita, M.; Pikuliak, M.; Srba, I.; Le, T.; Lee, D.; Simko, J.; et al. MULTITuDE: Large-Scale Multilingual Machine-Generated Text Detection Benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), Singapore*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 9960–9987. [CrossRef]
27. Dugan, L.; Hwang, A.; Trhlik, F.; Zhu, A.; Ludan, J.M.; Xu, H.; Ippolito, D.; Callison-Burch, C. RAID: A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; Volume 2024, pp. 12463–12492. [CrossRef]
28. Weber-Wulff, D.; Anohina-Naumeca, A.; Bjelobaba, S.; Foltynek, T.; Guerrero-Dib, J.; Popoola, O.; Sigut, P.; Waddington, L. Testing of Detection Tools for AI-Generated Text. *Int. J. Educ. Integr.* **2023**, *19*, 26. [CrossRef]
29. Wu, J.; Yang, S.; Zhan, R.; Yuan, Y.; Wong, D.F.; Chao, L.S. A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions. *arXiv* **2023**, arXiv:2310.14724.
30. Chakraborty, S.; Bedi, A.S.; Zhu, S.; An, B.; Manocha, D.; Huang, F. On the Possibilities of AI-Generated Text Detection. *arXiv* **2023**, arXiv:2304.04736.
31. Addanki, K.; Corrin, L. *Unveiling the Potential of Digital Twin Technology for Higher Education*; ASCILITE Publications: Tugun, Australia, 2023; pp. 12–21. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.