Advancing Machine-generated Text Detection: A Comprehensive Evaluation of Transformer-based Models

Batyr Sharimbayev [1,*] and Shirali Kadyrov [2]

[1] Department of Information Systems, SDU University, Kaskelen, Kazakhstan
[2] Department of General Education, New Uzbekistan University, Tashkent, Uzbekistan
Email: batyr.sharimbayev@sdu.edu.kz (B.S.); sh.kadyrov@newuu.uz (S.K.)
*Corresponding author

***Abstract*—Improved fluency in large language models has intensified the need for accurate detection of machine-generated text. This study evaluates transformer-based models using an improved version of the Conference on Computational Linguistics 2025 (COLING 2025), Generative Artificial Intelligence (GenAI) Content Detection Task 1 dataset, which was carefully preprocessed to enhance label quality and balance. All models were trained under a unified protocol to ensure fair comparison and robust evaluation. Test set results show that Decoding-Enhanced Bert with Disentangled Attention (DeBERTa) achieves the highest macro F1−Score of 85.48%, surpassing the previously top-ranked Multi-Task Learning (MTL) system, which attains a macro F1 of 83.07%. These results highlight the effectiveness of advanced transformer architectures for distinguishing human-written and machine-generated text. Despite these gains, performance degradation under domain shift and highly paraphrased inputs remains a challenge.**

***Keywords*—text classification, Artificial Intelligence (AI), AI-generated content, transformers, Decoding-Enhanced Bert with Disentangled Attention (DeBERTa), Artificial Intelligence (AI) detection, Bidirectional Long Short-Term Memory (BiLSTM)**

I. Introduction

The rapid progress of Large Language Models (LLMs) has made machine-written text more fluent, making it increasingly difficult to distinguish it from human-written text [1]. As technology has advanced, we need ways to tell the difference between text written by people and text written by machines based on certain features.

The traditional approaches for early detection were based on manually designed linguistic and statistical features. These approaches face challenges in generalizing on contemporary outputs produced by modern LLMs [2, 3]. Based on this requirement for effectiveness, transformer architecture models emerged at the forefront of contemporary research, showing excellent performance because of the ability of transformer models to perform contextual understanding [4, 5]. Fine-tuned models like Bidirectional Encoder Representations from Transformers (BERT), Robustly Optimized Bert Pretraining Approach (RoBERTa), and Decoding-Enhanced Bert with Disentangled Attention (DeBERTa) perform exceptionally on meticulously designed benchmark tasks [6]. However, robustness is still an issue that must be dealt with because detectors often perform sub optimally even when faced with out-of-domain data, adversarial paraphrasing techniques, and novel generative models [7, 8]. Researchers have been working on hybrid and ensemble models that use language features, recurrent neural networks, and transformer-based encoders [9, 10]. Still, the data distribution and variability [11] are still the key variables that affect performance. Generative models are changing quickly, so it's important to test them across a wide range of architectures to determine which ones work best for detection.

Systems for detecting Machine-Generated Text (MGT) are becoming more and more important in the real world in many different areas. In education, these tools can help teachers and universities find papers that were written with AI systems, which helps keep academic integrity [1]. In journalism and online media, detection models can help find fake reviews, automatically generated news, or misleading content that could influence public opinion [7]. These systems can also be used by social media sites to find spam, bots, or large-scale automated content generation. Furthermore, in research and publishing, detection tools can support editors in verifying the originality of submitted work. Because of these practical applications, it is important to develop detection models that are not only accurate, but also robust and reliable in real-world conditions where data can be diverse and constantly changing [11].

In our work, we conduct a systematic comparison of five transformer architectures: BERT, RoBERTa, DistilBERT, A Lite Bert (ALBERT), and DeBERTa. We use the Conference on Computational Linguistics 2025 (COLING 2025), Generative Artificial Intelligence

Manuscript received February 24, 2025; revised March 31, 2026; accepted April 14, 2026; published July 23, 2026.

(GenAI) Content Detection Task~1 dataset and analyze the accuracy and robustness to generalize on the development and test splits. Our aim is to find architectures that perform well while being robust to variation on data.

It is our hypothesis that the performance ceiling in today's benchmarking is often driven not by the model design, but by the quality of the data. The analysis of the original data suggests that most MGTs are "humanized" in some fashion by secondary tools. The MGTs can contain a lot of "noise" in the form of what are likely to be "humanized" texts that do not have the same "fingerprint" as the LLM outputs. However, we contend that the need to learn from such ambiguous data results in the learning of contradictory patterns, obscuring the decision boundary defined by human versus machine-written content. Aiming to mitigate such challenges, we propose a data-focused improvement approach. In our approach, we enhanced the COLING 2025 dataset to separate Generative Pre-trained Transformer (GPT) written content from noisy humanized texts that make learning harder. By focusing on high-quality data, we show that simpler transformer models perform better than some complex transformers. The previous top-ranked Metric Temporal Logic (MTL) system also shows that GPT-generated text can be detected very well on both development and test sets, which means this data is clean and suitable for training detection models.

There are three major contributions made by this paper. First, it provides a common framework of benchmarking that compares the performance of a BiLSTM baseline approach and five different transformer models in a realistic dataset of machine-generated texts. Second, it provides an analysis of the impact of different architectures in machine-generated texts, such as disentangled attention mechanisms and light-weight parameter-sharing, and their effect on the performance of the models for the detection of the machine-generated texts. Third, it provides an overview of the common limitations found in human and machine-generated texts, which provides insights into developing better models for the detection of machine-generated texts.

II. Literature Review

Researchers have employed various machine learning algorithms to identify text that has been created by the AI, including traditional statistical analysis and deeper learning algorithms that focus on transformers and LSTMs. In the early work, traditional algorithms used crafted features such as stylometry analysis and lexical statistical analysis. However, the nature of text generation is such that it needs deeper, semantic-level understanding, which can be offered by deep learning algorithms [2, 3, 7].

Transformers rely on self-attention mechanisms that allow models to view long dependencies between different points within the content. A series of works has proven that fine-tuned transformer models like RoBERTa, DistilBERT, or BERT are highly effective for distinguishing machine-generated content from human-written content. For instance, it has been shown that fine-tuned models have high accuracy, with values close to 0.99 for F1−Score, while being highly robust against adversarial attacks if well-trained [4−6]. Transformer models benefit greatly from diverse datasets because it decreases bias towards domains. However, many tasks remain to be addressed. Where a model might have done well in fine-tune tasks based on one dataset, they fail when used for out-of-domain tasks or adversarial paraphrasing's [7, 8]. By overcoming the limited capacity of current methods to learn deep discriminative features between human-written and AI-generated content, Tatavarthi *et al*. [12] hoped to improve machine-generated text detection. The authors present Generative Detection Generative Adversarial Network (GenDetect-GAN), a generative adversarial network-based model where a discriminator is iteratively trained to enhance detection capabilities while a generator learns the distribution of actual human text. According to experimental results, GenDetect-GAN performs better than current MGT detection models on a variety of evaluation metrics. Lee *et al.* [13] showed that the BL-CNN-BL model with fast text achieves high accuracy in news text classification, demonstrating its effectiveness for detecting fake news. The authors train and assess several machine learning models to detect generated reviews by combining probability-based sampling features from GPT-2 with conventional text mining features. With a macro F1−Score of 90% and up to 90% classification accuracy, the results demonstrate that the combined feature approach significantly outperforms text-only approaches.

Zero-shot detection techniques rely on a transformer model without any further fine-tuning. These techniques have proved to work effectively, especially in situations where there is a need for quick deployment. However, these techniques have proved to fail in situations where there is a need to identify subtle linguistic patterns. To overcome these challenges, researchers have opted to use techniques involving multiple transformer models. These techniques are designed to help overcome obstacles such as adversarial changes and paraphrasing [10, 14, 15].

Even with the current dominance of the transformer architecture, LSTM-based networks still retain some significance. LSTM-based models can efficiently pinpoint the dissimilarities in global coherence between texts produced by Artificial Intelligence (AI) and those produced by humans. They are designed to effectively identify sequential patterns in long sequences. Many research papers have attempted to integrate LSTM-based networks with other neural models or even with features extracted using sentence transformers to enhance the efficiency of the transformer architecture [9, 16]. These hybrid models take advantage of the detailed feature extraction capabilities of the transformer architecture and the sequential feature representation capabilities of the LSTM architecture. They show promising results in both binary and multiclass classification scenarios [6].

In addition to this, several researchers have also explored ensemble systems for improving the detection accuracy. Ensemble systems use the output of multiple models and include complex models such as the transformer model and Term Frequency-Inverse Document Frequency (TF-IDF)-based models, along with

traditional models such as random forest and Support Vector Machine (SVM). Ensemble systems generally have better detection accuracy and robustness by taking into account multiple views of the data. The models are generally better than traditional classifiers in terms of macro F1−Scores [10, 14, 15]. The models take into account several factors such as language statistics, lexical diversity metrics, and readability scores. In addition to this, purely feature-based models such as those based on linguistic indicators and statistical patterns such as n-gram statistics and perplexity are also important. Such models are important when combined with deep representation and form a strong foundation for developing detection systems [2, 17].

The importance of the characteristics of the dataset in the performance of detection has been discussed. It has been argued in several studies that for achieving high generalization performance between different models, it is essential to have high diversity in the dataset. Overfitting and out-of-domain performance can happen due to the models that have been trained on limited examples, which have not been exposed to sufficient representation of diversity in use cases in MGT. In fact, it has been found in several research studies that the sensitivity of models to adversarial attacks and paraphrasing can depend heavily on the composition of the dataset [2, 7]. The use of techniques such as adversarial training has been found to be very effective in countering these issues without affecting the accuracy of detection [11, 18].

However, robustness remains one of the biggest challenges for detection systems. While detection systems have shown promising results in terms of accuracy, many detection models have shown to be vulnerable to paraphrasing and other attempts to avoid detection. For instance, transformer models have shown to struggle in maintaining performance when the input text is paraphrased or when there is an addition of adversarial noise to the input text [5, 7, 8]. Recent studies have shown that hybrid models can help in increasing robustness to input changes [4, 6].

Research on the detection of AI-written texts can be grouped into a few categories. Popular models like BERT and RoBERTa are very effective because they understand the meaning of the sentences. Nevertheless, these models can be challenged when the text is on a new topic or if it is manipulated to evade detection. For this reason, some models have been developed to integrate different designs to ensure the model understands both the meaning and the word order. Even though these models are improved versions, they are complex and require a lot of computation. Other models rely on grammar and statistics. These models are easy to understand but lack the deep understanding required. Finally, some approaches use multiple models together to get the best results, although this can be slow and expensive. Because of these issues, there is still a need for detection tools that are simple, reliable, and work well with many different types of data.

III. Data Preparation

A. Data Collection

We used the English-only dataset from Subtask A of the GenAI Content Detection Task 1 released at the COLING 2025 Workshop [19]. The corpus contains human-written texts and MGT from multiple LLMs. Following prior findings indicating that GPT-generated texts are more consistently detectable than outputs from other models [20], we restricted the training and development sets to GPT-based generations, while retaining a broader range of models in the test set to evaluate generalization.

To establish a clearer decision boundary, we applied a model targeted filtering strategy by restricting the machine generated data in the training and development sets to GPT family models, including GPT-3.5-turbo, GPT4o, GPT-4o-mini-2024-07-18, GPT-4o-2024-08-06, GPT-35, GPT4. We conducted strategic subsampling by removing samples from non GPT generators in order to reduce conflicting patterns and improve the clarity of the human versus machine distinction during the early stages of model learning.

The final dataset contains 235,192 samples, which are part of the original set. We split this set into 188,398 training samples and 47,100 development samples. We do not modify the test set; it still includes all 73,941 samples from the original Task 1. This is to test whether our models can work well with “noisy” or “human-like” text that we have not seen before. More information about the split for each set is provided in Table.

B. Data Normalization

To guarantee the consistency of these labels, we standardized the model names in the data set first. The original data set uses different names for the same GPT models. To correct this problem, we created a list to map these different names and replace them with a standard title. This ensures the model will not misinterpret by different names. Finally, we merged the training set and development set into one set based on the official rules of the task.

Although these processes of cleaning data enhance the quality of the labels, there are some limitations as well. Since our models are trained on data generated by GPT, there is a possibility that these models might not recognize text generated by other types of AI systems. There is also a possibility that different versions of a model may bear different styles of writing; therefore, by giving different versions a single name, there is a possibility that some crucial information might be lost. To avoid these limitations, we are using a diverse test set consisting of various models in order to check how well our system performs.

C. Data Exploration

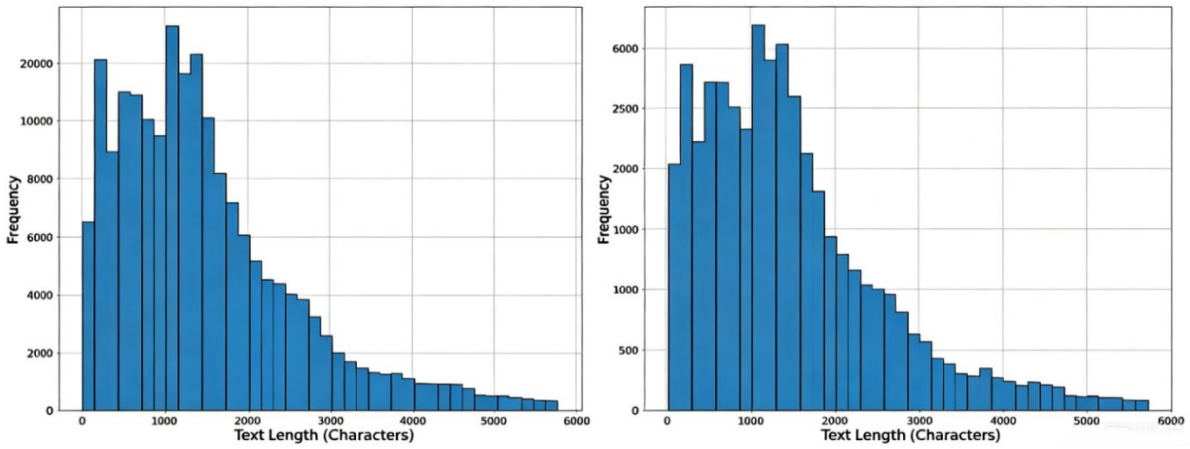
To analyze the lexical difference in terms of vocabulary used in human-written texts and AI-generated texts, a word frequency analysis of the cleaned data in the training set was conducted (see Fig. 1). As can be seen, the vocabulary used in MGTs is dominated by functional words such as make, use, help, important, need, etc., which illustrates the

formal tone and sometimes academic tone used by GPT. On the contrary, in human-written texts, a range of experiential words are used, including time, people, even, work, way, etc.

We further inspected the distribution of text lengths for the training and development sets after the removal of the extreme outliers above the 97.5th percentile (Fig. 2). The training distribution is right-skewed and unimodal, and the vast majority of the texts lie in the range from 1000 to 1500 characters, indicating more variability in human-written texts. The distribution for the development set is more distinct and bimodal, peaking at both 1800 and 2900 characters. The length of GPT-generated texts is presumably set accordingly. Nevertheless, the core regions for both distributions overlap significantly, thus diminishing the distribution difference.

(a) (b)

Fig. 1. Comparison of word frequency distributions between human-written texts and Machine-Generated Texts (mgts): (a) word cloud for mgts, (b) word cloud for human-written texts.



(a) (b)

Fig. 2. Text length distributions filtered at the 97.5th percentile: (a) Training set, (b) Development set.

IV. Model Selection

For designing an effective detection system capable of differentiating human-written text from MGT, the need to incorporate architectural diversity as part of the model selection approach arose. With the accelerated development of LLMs and the complexity of generated text increasing continuously, it is imperative to consider models that address various linguistic aspects.

Our selection criteria were based on three main goals: (1) comparing architectures with fundamentally different designs to identify their unique strengths, (2) including both traditional and advanced models, and (3) evaluating models widely used in current text classification research. We chose ALBERT and DistilBERT because we were interested in exploring how sharing parameters can impact the performance of a text classifier in determining AI-generated text. We chose BERT as a baseline model for our experiments. We were interested in exploring how better training methods can impact performance, so we chose RoBERTa as another model in our experiments. Finally, we chose DeBERTa to explore how better attention can impact performance in determining patterns in AI writing.

A. Bidirectional-LSTM

For establishing a basic baseline, we chose to incorporate a Bidirectional Long Short-Term Memory (Bi-LSTM) network. Bi-LSTM networks belong to the class of traditional recurrent neural networks and are designed for investigating the relationships in sequential data over large periods [21−23]. Although the transformer architecture has received considerable attention in modern studies, the network can still act as a useful baseline because of its capacity for analyzing the flow and sequential structure in stories. The existing studies suggest

the capacity of the networks for competitively identifying the authored and generated parts.

The implemented model employs a two-layer Bi-LSTM architecture, with each layer containing 128 hidden units. Input texts are first mapped to dense representations using an embedding layer with a vocabulary size of 10,000 and an embedding dimension of 256. The first Bi-LSTM layer outputs full token-level sequences, while the second produces a final sequence representation. This is followed by a fully connected layer with 64 units, where the activation function is ReLU, and the dropout value of 0.3 has been used to prevent overfitting. In the final output layer, the sigmoid activation function was employed since it is a binary classification task. The model has been trained using the Adam optimizer with a learning rate of 1×10^{-4}. The final training has been performed on the basis of the optimized loss function of binary cross-entropy. All input sequences are padded or truncated at a maximum length of 257 tokens to keep the input dimensions uniform. Comprehensive experiments were performed on different combinations of values, and finally, the values were chosen on the basis of their robust performance.

B. Transformer-Based Models

Transformers are the current state-of-the-art models in natural language processing tasks owing to their effective modeling of context using self-attention mechanisms. To allow a fair comparison of models, we chose a variety of transformer models that include lean models that consume low resources and more complex models that are state-of-the-art (Table I). These models give us an insight into how architectural choices influence their predictive accuracy on our specific problem of distinguishing text created by a human compared to a GPT model.

TABLE I. DESCRIPTIONS AND MOTIVATIONS FOR TRANSFORMER MODELS INCLUDED IN THE EVALUATION.

No.	Model	Motivation	Description
1	DistilBERT [24]	Efficiency baseline and parameter efficiency	A compressed variant of BERT that retains much of its representational capacity while significantly reducing computational overhead and an efficiency transformer baseline.
2	BERT [25]	Standard benchmark	A canonical bidirectional encoder that forms the foundation of modern NLP research and provides a standard benchmark for comparison.
3	RoBERTa [26]	Robust optimization	An improved and robustly optimized variant of BERT trained on larger corpora with enhanced objectives, often demonstrating stronger generalization.
4	ALBERT [27]	Efficiency baseline and parameter efficiency	A parameter-efficient architecture employing cross-layer parameter sharing and factorized embeddings, enabling lightweight transformer deployment under resource constraints.
5	DeBERTa-v3 [28]	SOTA performance for binary text classification	A modern architecture using disentangled attention and enhanced position encoding, exhibiting strong semantic representation for subtle stylistic discrimination.

We had a clear process to fine-tune these models. For fine-tuning, we used specific tokenizers and set the length of these models to 512 tokens. The learning rate for these models was set to the optimal value, ranging from 1×10^{-5} to 1×10^{-4}, and the models were trained for 1 to 6 rounds, as shown in Table. Other important parameters, like batch size and weight decay, were set to ensure that the models were learning correctly and not becoming unstable. The performance of these models was then calculated in terms of accuracy and F1−Score, which is the standard method to test the models for the COLING 2025 competition.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section will display the performance results of all the models we have tested. The evaluation of the models consists of five transformer models, and we have used the Bi-LSTM model for the development set and the best MTL system from the COLING 2025 competition for the test set. The best decision threshold was determined by testing different values in the range [0.1, 0.95] of the decision threshold on the development set to get the best F1−Score. After determining the best decision threshold, we used it to get the results for both the development and test sets.

A. Development Set Results

The results for the development data in Table II indicate that, in general, the performance of the transformer models is better than that of the baseline Bi-LSTM model. The most significant result, however, is the excellent performance of the DeBERTa model, which attained a high Macro F1 of 98.15%. The achievement of this performance by the model is even more significant because it attained this performance at a very high threshold of 0.95. This implies that the model is very confident in its predictions. The advanced architecture of the model allows it to detect small patterns in the way the AI model generates sentences, even though the system is very cautious.

TABLE II. MODEL PERFORMANCE ON THE DEVELOPMENT SET SORTED BY MACRO F1−SCORE (%)

No.	Model	Threshold	Accuracy	Macro F1
1	DeBERTa	0.95	98.15	98.15
2	DistilBERT	0.95	97.73	97.73
3	ALBERT	0.52	97.16	97.16
4	BERT	0.69	96.51	96.50
5	Bi-LSTM (baseline)	0.62	95.83	95.83
6	RoBERTa	0.74	93.64	93.64

However, there is another surprising trend concerning smaller models, and DistilBERT and ALBERT performed better than standard BERT and RoBERTa models. The score of DistilBERT was 97.73%, and the score of ALBERT was 97.16%. This might be because these models were created to be fast and use less memory. This means that, in this specific case, smaller models might be better at concentrating only on the most important features of the text and not incorporating irrelevant noise.

The classification thresholds also indicate the reliability of each model. For instance, DeBERTa and DistilBERT were extremely reliable even when the threshold was set to 0.95. However, in the case of the ALBERT and

Bi-LSTM models, the thresholds were set to 0.50 to ensure the models performed optimally. This means that DeBERTa and DistilBERT can be used safely, even in real-world scenarios, because the high threshold ensures that human writers are not accused of using AI. Lastly, although the Bi-LSTM model performed well, achieving 95.83%, it is a simple model and might not perform well in new and complex AI writing styles compared to the transformer models. It is important to note that the lowest score was achieved by the RoBERTa model, which was 93.64%.

B. Test Set Results

From Table III, it is clear that the results confirm that the most effective model for the task is DeBERTa, which obtained a Macro F1 of 85.48%. Just like in the development phase, the model obtained the results by utilizing a high threshold of 0.95. This shows that not only is the system accurate but also very confident in identifying a text as being generated by an AI. The success of the model also shows that advanced attention mechanisms are very effective in identifying Machine-Generated Text (MGT), even when working with new data.

TABLE III. MODEL PERFORMANCE ON THE TEST SET SORTED BY MACRO F1−SCORE (%)

No.	Model	Threshold	Accuracy	Macro F1
1	DeBERTa	0.95	85.48	85.48
2	RoBERTa	0.74	83.94	83.93
3	DistilBERT	0.95	83.65	83.62
4	MTL (Top 1) [20]	0.95	83.11	83.07
5	ALBERT	0.52	82.75	82.74
6	BERT	0.69	78.80	78.76
7	Bi-LSTM (baseline)	0.62	73.65	73.52

The greatest achievement of this research is that DeBERTa performed better than the previous best system, which was referred to as the MTL model system. The MTL system scored 83.07%, but DeBERTa managed to surpass this by more than two percent. This achievement demonstrates the advantages of using a transformer architecture in recognizing complex patterns in text written by large language models.

The results for the other models also reveal some interesting findings. For instance, the performance of the RoBERTa and DistilBERT models was not far behind the best-performing model, achieving 83.93% and 83.62%, respectively. What is more interesting is that the DistilBERT model, which is a smaller and more efficient model, managed to remain very competitive even with a very high 0.95 threshold. Another surprise in the results is the performance of the ALBERT model, which achieved 82.74% compared to the performance of the standard BERT model. This reveals that perhaps the efficiency features of these models help them concentrate more on the most useful linguistic features for this specific dataset.

Finally, the Bi-LSTM baseline model performed the worst, with a score of 73.52%. It is evident that there is a significant disparity between the baseline model and the transformer models. This is proof that the attention-based model is much better at dealing with complex text classification problems. It is also evident that the different thresholds, such as 0.74 for RoBERTa and 0.52 for ALBERT, imply different levels of confidence for each model. Nevertheless, the models with high accuracy at the 0.95 threshold remain the best to rely on, especially when avoiding false accusations is a priority.

C. Statistical Analysis

To verify if the difference in the performance of the models is statistically significant, two different statistical tests were carried out. First, Cochran's Q test was conducted to verify if there is a general difference in the performance of all six models. The test verifies if all models have the same performance or if at least one of them is significantly different from the others. The test results (Q = 2,659,252.50, $p < 0.001$) reject the null hypothesis very strongly, verifying that the models do not actually have the same performance.

After that, we compared the models pairwise with the McNemar test. This test can be used to compare two models based on the same data set by examining the cases in which predictions made by the models differ. To ensure the results were reliable for multiple comparisons, we used the Benjamini and Hochberg correction. All results were still significant after correction ($p_{corrected} < 0.001$). For the sake of conciseness, Table IV only compares the best model, DeBERTa, with the other models. In Table IV, n_{10} denotes the number of cases in which DeBERTa was correct and the other model was incorrect, and the opposite applies for n_{01}. The χ^2 values and corrected p-values in Table IV suggest that all other models perform significantly worse than DeBERTa.

This data makes a strong case for the efficacy of the DeBERTa model, which manages to beat the performance of other models such as RoBERTa, DistilBERT, ALBERT, BERT, and Bi-LSTM. The high number of discordant pairs, which is the combination of n_{10} and n_{01}, and the low p-value are not anomalies in the way the math is working out; it is the actual performance gap in the huge data set.

TABLE IV. POST-HOC MCNEMAR TESTS (BH-CORRECTED) COMPARING DEBERTA WITH OTHER MODELS ON THE TEST SET

No.	Model	Accuracy (%)	Δ Accuracy	n_{10}	n_{01}	χ^2	$P_{corrected}$
1	RoBERTa	83.94	−1.54	2788	3928	193.17	<0.001
2	DistilBERT	83.65	−1.83	2956	4308	251.27	<0.001
3	ALBERT	82.75	−2.73	2736	4760	545.96	<0.001
4	BERT	78.80	−6.68	1926	6865	2773.73	<0.001
5	Bi-LSTM	73.65	−11.83	1.0673	1921	6080.67	<0.001

Note: Accuracy denotes accuracy (%), and ΔAccuracy indicates the difference relative to DeBERTa. n_{10} denotes instances correctly classified by DeBERTa but misclassified by the compared model, while n_{01} denotes the reverse. χ^2 denotes the McNemar test statistic. $P_{corrected}$ are reported after Benjamini–Hochberg correction.

When comparing these figures to the previous MTL system, the accuracy of the new system at 83.11% means that its 95% confidence interval falls between 82.84% and 83.38%. Comparing this to DeBERTa's interval of 85.4% to 85.9%, it is clear why there is an improvement. Of course, in order to draw a definitive statistical-based conclusion, we would need access to the raw data in order to compare the two on a case-by-case level.

D. Error Analysis

Fig. 3 demonstrates the varying detection accuracy based on the source of the text. The most advanced models, like GPT-4o, GPT-4o-mini, and LLaMA-3.1, have detection rates of over 98% in almost all cases, indicating that the writing style still bears a highly recognizable mark. Text written by humans is detected reliably with a 93.6% rate.

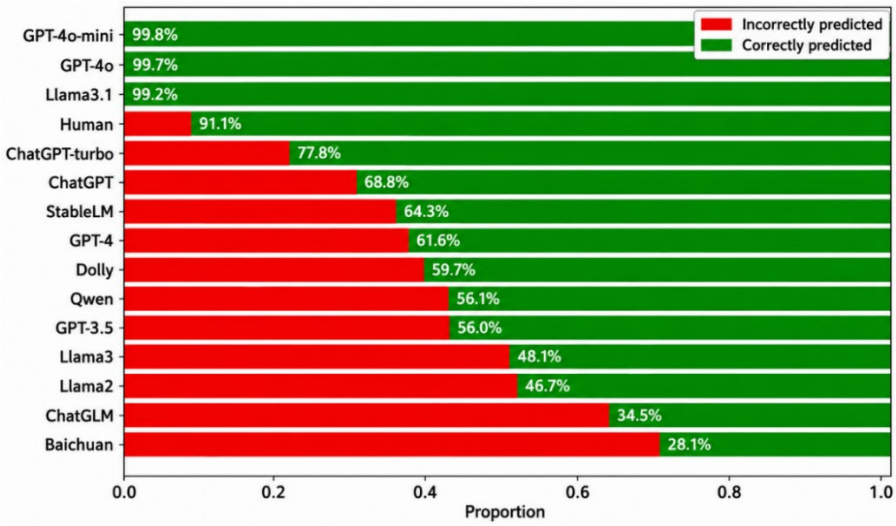


Fig. 3. Prediction accuracy and error distribution across human-written text and different generative models.

However, this accuracy is significantly reduced for older or different versions. For example, the accuracy for GPT-3.5 is only 49.7%, while the accuracy for Baichuan and Chat General Language Model (ChatGLM) is the lowest, at 28.1% and 34.5%, respectively. This low accuracy suggests that there is a significant stylistic similarity between these generators and natural human speech. The basic reason is that as AI-generated text becomes less formulaic, or as we say, "humanized" through post-processing, the transformers' ability to distinguish between such texts and natural human writing is significantly reduced.

The source of the dataset, i.e., the provenance, plays an important role in the frequency of detection error occurrence, as shown in Fig. 4. For the structured benchmarks like NLPeer-ARR22, NLPeer-COLING20, and peersum, the detection accuracy is almost perfect, indicating that the gap between human and machine writing styles might be quite sharp. However, the "applied" datasets like Mixset, CUDRT, and IELTS-related datasets are more difficult. The main reason for this difficulty is that the responses in these datasets might be short, formulaic, or restricted to specific topics. This means that human and AI writing styles might be quite similar, making it difficult to find distinguishing signals.

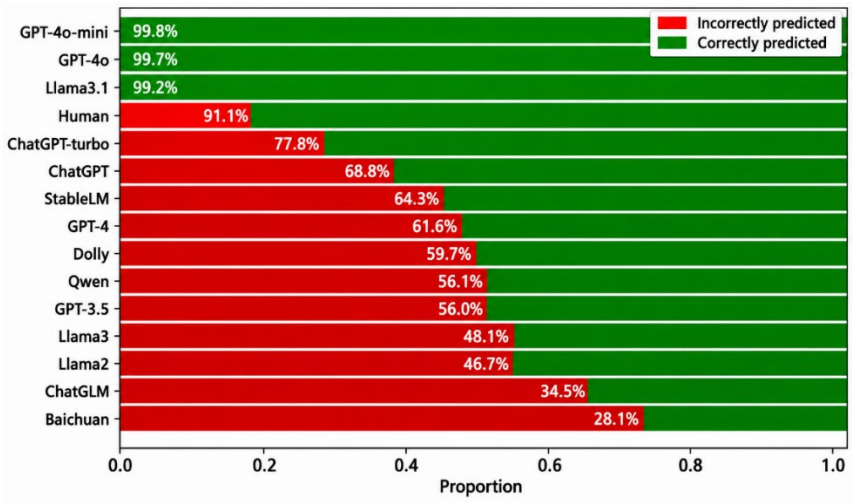


Fig. 4. Detection accuracy and error rates across different dataset sources

TABLE V. ERROR-FOCUSED ANALYSIS OF MODEL PREDICTIONS BY TEST SET

Source	Num. Errors	FP Total	FN Total	Mean Length	Median Length	Mean TTR	Mean Confidence
CUDRT	6,707	60	6,647	239	251	0.649	22.5
LLM-DetectAIve-IELTS	272	255	17	302	299	0.569	21.9
Mixset	1,790	154	1,636	129	127	0.737	19.6
NLPeer-ARR22	1	0	1	210	210	0.700	19.1
NLPeer-COLING20	1	1	0	275	275	0.520	13.8
NLPeer-F1000	124	93	31	327	299	0.568	21.2
ieltsduck	1,724	1.712	12	302	297	0.570	21.8
peersum	115	54	61	218	234	0.632	25.3

Table V identifies the error patterns based on different test sources and provides a clear view of where exactly the predictions are diverging. For instance, high FP rates are seen in the "ieltsduck" and "LLM-DetectAIve-IELTS" datasets, which indicates that the model is over-identifying human-written content as machine-written. On the other hand, datasets such as "CUDRT" and "Mixset" have a high number of False Negatives (FN), where the model is over-accepting machine-written content as human-written. Analyzing these errors indicates that longer and lexically diverse texts, based on the Type-Token Ratio (TTR), are being misclassified. Moreover, the over-confidence of the model in its false positive predictions provides a better view of why the model is having difficulty with these complex inputs.

Further examination of the data set showed that there were a number of incorrect labels in the data set. The label "0" was supposed to be for entirely human-written texts, but in fact, it contained 600 texts created by AI models, including Dolly (144), StableLM (136), and GPT. The labeling was incorrect, and this is important because it means that many of the model's supposed "mistakes" were in fact cases in which it correctly identified texts created by AI but which had been incorrectly labeled as human-written. It should be noted, however, that 79.1% of these incorrect classifications were contained in the ieltsduck and LLM-DetectAIve-IELTS data sets. This can be explained by the fact that IELTS essays have a highly rigid structure. Because both human students and AI models must follow the same rigid structure, formal vocabulary, and standard transitions in order to fulfill the requirements of the exam, the style of human and artificial writing becomes almost identical.

The confusion matrix in Fig. 5 reveals a clear imbalance in error types: false negatives occur more frequently than false positives. Many machine-generated texts, particularly longer and well-structured outputs, are predicted as human, indicating a conservative bias that favors avoiding false accusations. False positives are less common and typically involve concise, formal, or information-dense human texts resembling instructional AI style.

This is because of a deliberately cost-sensitive configuration, where a classification threshold was set at 0.95. The consistency of the probability values provided can be seen in the Calibration Curve in Fig. 6, where a comparison of predicted confidence and actual proportion of machine-generated text is shown. The system has deliberately set a high threshold to ensure that human authors are not misclassified, even if it means more false negatives. Ultimately, these errors are not random; they are a result of the convergence of human and machine styles.

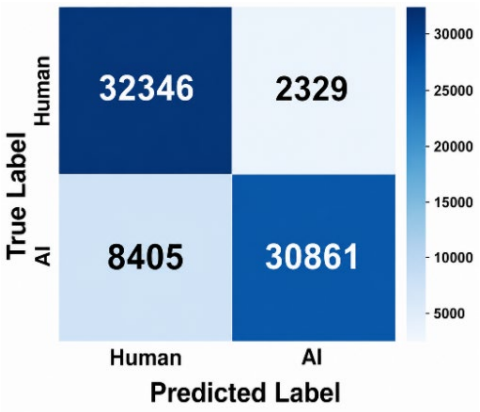


Fig. 5. Confusion matrix of the DeBERTa model.

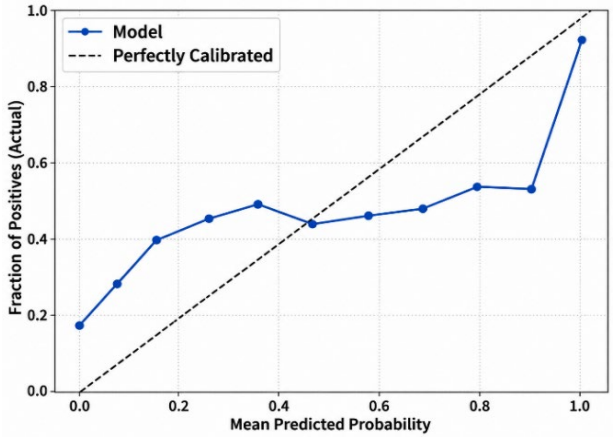


Fig. 6. Calibration analysis of the DeBERTa model.

VI. CONCLUSION

The results indicate a distinction in performance between recurrent and transformer-based models in identifying MGT. DeBERTa performed best, primarily because of its architecture. It is widely regarded as a top-performing model for text classification. One of its key strengths is a feature known as "disentangled attention", whereby a word's meaning and position in a sentence are treated as separate information inputs. This enables a far more effective understanding of relationships between words compared to earlier versions of such models. Consequently, DeBERTa is able to recognize subtle patterns in MGT that other models might not be able to recognize. In addition, its more effective approach to understanding word positions enables it to maintain a high

level of accuracy even when confronted with complex text. Nevertheless, it is worth noting that these findings were based on a research model that was specifically focused on GPT-generated text. While such a focused approach enables a more precise level of detection within a particular field, it also leads to an awareness that such findings might not be applicable to MGT generated through other AI tools. Thus, although transformer-based models perform highly effectively in such specific areas, their performance in relation to a broader range of AI tools is an area that warrants further research. Future work should include broader training mixtures and targeted ablation studies to better isolate these factors.

From the analysis of the mistakes, it was observed that it is still difficult to differentiate between clear machine-generated texts and human-written texts. It was also observed that short human texts were sometimes mistaken for being machine-generated. This indicates that the mistakes in detection are not entirely random, but rather because the styles of writing are becoming more and more similar. Hence, it can be concluded that transformer-based models can be used in real-world scenarios, but the challenge remains in ensuring proper functioning with varying topics and styles.

In our future research, we plan to use ensemble models, which will combine the benefits of different models to produce better results. Hybrid models that use a combination of the transformer and other neural network models may also be better in recognizing both local and global patterns in the text. The use of adversarial training, which trains the detector on text that is meant to deceive it, could also improve the reliability of the detector and enable it to work better even if it is faced with new AI models that it is not aware of.

APPENDIX: FINAL SYSTEM HYPERPARAMETERS FOR ALL MODELS

Experiments were conducted on NVIDIA A100 Tensor Core GPUs using Google Colab. See hyperparameters in Table AI. Table AII presents the statistics of the dataset splits. The training set contains 188,398 samples, with a relatively balanced distribution between generated (G) and human-written (H) texts. The average and median text lengths indicate that generated texts tend to be slightly longer than human-written texts across all splits, particularly in the test set.

TABLE AI. FINAL HYPERPARAMETERS OF EVALUATED MODELS

No.	Model	Learning rate	Warmup ratio	Weight decay	Batch size	Epoch	Average Epoch Time (s)
1	DeBERTa	1×10^{-5}	0.15	0.01	32	1	3,051
2	BERT	2×10^{-5}	0.1	0.01	16	2	2,625
3	RoBERTa	1×10^{-5}	0.1	0.1	16	3	3,058
4	DistilBERT	5×10^{-5}	0.06	0.01	32	3	1,290
5	ALBERT	1×10^{-5}	0.1	0.01	16	3	2,340
6	Bi-LSTM	1×10^{-4}	–	0.01	32	6	2.72

TABLE AII. STATISTICS OF THE DATASET SPLITS (G = MACHINE-GENERATED, H = HUMAN-WRITTEN TEXT)

Split	#Samples	M / H	Mean Length (M / H)	Median Length (M / H)
Train	188,398	91,935 / 96,463	296 / 267	249 / 181
Dev	47,100	23,064 / 24,036	294 / 265	249 / 181
Test	73,941	39,266 / 34,675	357 / 244	383 / 261

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Batyr Sharimbayev conducted the research, performed the experiments, and wrote the manuscript; Shirali Kadyrov supervised the study, provided guidance on methodology, and reviewed the manuscript; all authors have read and approved the final version of the manuscript.

FUNDING

This research was funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan within the framework of project AP23487777.

ACKNOWLEDGMENT

The authors wish to thank the COLING 2025 organizers for providing the competition and dataset.

REFERENCES

[1] A. Amirzhanov, C. Turan, and A. Makhmutova, "Plagiarism types and detection methods: A systematic survey of algorithms in text analysis," *Frontiers in Computer Science*, vol. 7, 1504725, 2025.

[2] E. N. Crothers, N. Japkowicz, and H. L. Viktor, "Machine-generated text: A comprehensive survey of threat models and detection methods," *IEEE Access*, vol. 11, pp. 70977–71002, 2023.

[3] S. D. Aldeen, T. Abbas, and A. R. Abbas, "Review of detecting text generated by ChatGPT using machine and deep-learning models: A tools and methods analysis," *Diyala Journal of Engineering Sciences*, vol. 18, no. 1, pp. 34–54, 2025.

[4] M. M. Oghaz, L. B. Saheer, K. Dhame, and G. Singaram, "Detection and classification of ChatGPT-generated content using deep transformer models," *Frontiers in Artificial Intelligence*, vol. 8, 1458707, 2025.

[5] S. Abdali, R. Anarfi, C. J. Barberan, and J. He, "Decoding the AI pen: Techniques and challenges in detecting AI-generated text," in *Proc. 30th ACM Conf. Knowledge Discovery and Data Mining*, 2024, pp. 6428–6436.

[6] H. Abburi, S. Bhattacharya, E. Bowen, and N. Pudota, "AI-generated text detection: A multifaceted approach to binary and multiclass classification," arXiv preprint arXiv:2505.11550, 2025.

[7] J. Wu, S. Yang, R. Zhan, Y. Yuan, D. F. Wong, and L. S. Chao, "A survey on LLM-generated text detection: Necessity, methods, and

future directions," *Computational Linguistics*, vol. 51, no. 1, pp. 275–338, 2025.
[8] A. Makhmutova, B. Sharimbayev, A. Amirzhanov, and A. Shalkarbay-uly, "Testing the limits: Evaluating AI detectors' accuracy and the impact of obfuscation techniques on AI-generated text," *Journal of Advances in Information Technology*, vol. 17, no. 3, pp. 438–449, 2026.
[9] P. Petropoulos and V. Petropoulos, "RoBERTa and Bi-LSTM for human vs AI generated text detection," in *Proc. CLEF 2024: Conference and Labs of the Evaluation Forum*, 2024, pp. 2838–2842.
[10] H. Abburi, M. Suesserman, N. Pudota, B. Veeramani, E. Bowen, and S. Bhattacharya, "Generative AI text classification using ensemble LLM approaches," arXiv preprint arXiv:2309.07755, 2023.
[11] D. Valiaiev, "Detection of machine-generated text: literature survey," arXiv preprint arXiv:2402.01642, 2024.
[12] A. P. Tatavarthi, F. Abri, and N. Attar, "AI-generated text detection and source identification," *Journal of Advances in Information Technology*, vol. 16, no. 7, pp. 1030–1041, 2025.
[13] S. C. Lee, D. G. Lee, and Y. S. Seo, "Determining the best feature combination through text and probabilistic feature analysis for GPT-2-based mobile app review detection," *Appl. Intell.*, vol. 54, no. 2, pp. 1219–1246, 2024.
[14] M. K. Sheykhlan, S. K. Abdoljabbar, and M. N. Mahmoudabad, "Team karami-kheiri at PAN: Enhancing machine-generated text detection with ensemble learning based on transformer models," in *Proc. CLEF 2024: Conference and Labs of the Evaluation Forum*, 2024, pp. 2682–2687.
[15] S. Pudasaini, L. M. Pechuán, D. Lillis, and M. L. Salvador, "Enhancing AI text detection with frozen pretrained encoders and ensemble learning," in *Proc. CLEF 2025 Working Notes*, 2025, vol. 4038, pp. 3608–3614
[16] N. Islam, D. Sutradhar, H. Noor, J. T. Raya, M. T. Maisha, and D. M. Farid, "Distinguishing human generated text from ChatGPT generated text using machine learning," arXiv preprint arXiv:2306.01761, 2023.
[17] Z. Chen and H. Liu, "Stadee: Statistics-based deep detection of machine generated text," in *Proc. Int. Conf. Intelligent Computing*, 2023, vol. 14089, pp. 732–743.
[18] L. Lin, N. Gupta, Y. Zhang, H. Ren, C. H. Liu, F. Ding, X. Wang, X. Li, L. Verdoliva, and S. Hu, "Detecting multimedia generated by large AI models: A survey," arXiv preprint arXiv:2306.01761, 2023.
[19] Y. Wang *et al*., "GenAI content detection task 1: English and multilingual machine-generated text detection: AI vs. human," in *Proc. 1st Workshop on GenAI Content Detection (GenAIDetect)*, 2025, pp. 244–261.
[20] G. Gritsai, A. Voznuyk, I. Khabutdinov, and A. Grabovoy, "Advacheck at GenAI detection task 1: AI detection powered by domain-aware multi-tasking," in *Proc. 1st Workshop on GenAI Content Detection (GenAIDetect)*, 2025, pp. 236–243.
[21] B. Sharimbayev and S. Kadyrov, "Automatic language identification from audio signals using LSTM-RNN," in *Proc. 2023 17th Int. Conf. Electronics Computer and Computation (ICECCO)*, 2023, pp. 1–5.
[22] B. Sharimbayev and S. Kadyrov, "Text classification for AI generated content with machine learning and deep learning models," in *Proc. 2025 IEEE 5th Int. Conf. Smart Information Systems and Technologies (SIST)*, 2025, pp. 1–5.
[23] B. Sharimbayev, A. Kazin, and S. Kadyrov, "Multi-class detection of humanized AI text using machine learning and transformer models," in *Proc. 2025 Int. Conf. Computational Intelligence and Knowledge Economy (ICCIKE)*, 2025, pp. 127–131.
[24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019.
[25] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Am. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol*., vol. 1, 2019, pp. 4171–4186.
[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint arXiv:1907.11692, 2019.
[27] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," arXiv preprint arXiv:1909.11942, 2019.
[28] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," arXiv preprint arXiv:2006.03654, 2020.