CHAPTER «ENGINEERING SCIENCES»

IMPROVING DETECTION OF AI-GENERATED TEXT IN EDUCATION

Kateryna Antipova[1]
Hlib Horban[2]

DOI: https://doi.org/10.30525/978-9934-26-562-4-1

Abstract. Language models can paraphrase and create text that is almost indistinguishable from text written by humans. The emergence of such generative AI tools and their ability to generate human-like text poses a significant threat to academic integrity. While AI-generated general content might exhibit noticeable inconsistencies in the broader context, the narrower and more structured scope of academic assignments may mask such anomalies, making the detection process more complex. Hence, distinguishing AI-generated homework requires more refined and context-aware algorithms. Existing research has shown that text-matching software not only does not find all plagiarism, but might also incorrectly label highly formal or technical human writing as AI-generated, thus providing false positive results. The detection tools present a bias towards classifying the output as AI-generated rather than human-written. *The purpose of the paper* is to improve the accuracy and reliability of detecting AI-generated text, especially in the educational environment, where plagiarism and academic dishonesty are becoming increasingly relevant due to the use of generative language models. The study aims to adapt modern plagiarism detection methods for reliable classification of AI-generated texts in the context of the Ukrainian language. *Methodology of the study* is based on general research methods of analysis and synthesis, experimental testing, and

[1] Doctor of Philosophy,
Associate Professor at the Department of Software Engineering,
Petro Mohyla Black Sea National University, Ukraine
[2] Candidate of Technical Sciences, Associate Professor,
Associate Professor at the Department of Software Engineering,
Petro Mohyla Black Sea National University, Ukraine

© Kateryna Antipova, Hlib Horban

quantitative analysis to comprehensively examine and compare the efficacy and performance of different detectors utilized in machine-generated text detection. *The obtained results* show that the fine-tuned model effectively detects differences between the two types of text. The obtained results also provide some insight into the strengths and weaknesses of the model, and demonstrate its potential for application to practical tasks. Future research could explore the application of fine-tuned models to other languages and diverse content types. Besides that, expanding the dataset to include various styles and contexts will allow for a more robust evaluation of the model's performance. *Practical implications.* By adapting modern plagiarism detection methods, the research will contribute to the development of reliable tools that uphold academic integrity and prevent misconduct in student submissions. This study will help educational institutions implement more effective plagiarism detection systems that can accurately differentiate between human-written and AI-generated texts in Ukrainian. Our findings can also guide educators in creating fair and transparent regulations for the use of AI-generated content in academic settings. *Value/originality.* The scientific novelty of the study involves the adaptation of modern methods of plagiarism detection for reliable classification of texts created by artificial intelligence in the context of the Ukrainian language. For this purpose, we have created a new dataset based on paraphrased text fragments generated by ChatGPT. The efficiency of the fine-tuned model was evaluated using different evaluation metrics: accuracy, F1 score, true positive rate, and true negative rate.

1. Introduction

The introduction of ChatGPT, a revolutionary tool based on a large-scale language model (LLM), has significantly changed various industries and fields, including academia. The capabilities of advanced LLMs have impacted the academic world in a variety of ways. For example, higher education students are using ChatGPT to do their homework and take exams. This has raised concerns about the current assessment systems used in higher education institutions. Teachers and universities are trying to detect fraudulent activities of students, and plagiarism is one of the main problems. In the past, plagiarism mostly consisted of submitting papers and essays that contained paragraphs from other sources without citing them, but with the advent of LLMs, students can now use artificial intelligence (AI) to create text and complete their assignments. The act of students using text generated by a LLM and claiming it as their own work is called AI plagiarism. Students' dependence on text generation tools leads to a loss of creativity and learning ability.

Language models are deep machine learning-based models designed for various natural language processing (NLP) tasks. LLMs such as ChatGPT can create and paraphrase text that is almost indistinguishable from text written by humans. These models can handle both simple tasks, such as creating an essay on a given topic, and complex ones, such as writing a research paper on a complex problem. The emergence of such generative AI tools and their ability to generate human-like text poses a significant threat to academic integrity. Reliance on AI-generated homework may impede students from understanding their coursework, consequently undermining the educational experience. If students receive credit for other people’s work, then an important extrinsic motivation for acquiring knowledge and competences is reduced. Likewise, the assessment of students' competence is distorted, which can result in undue benefits for plagiarists.

The task of detecting whether a particular text is an AI generated text (AIGT) or a human written text (HWT) is called artificial intelligence content detection. With the predicted rapid development of high-performance LLMs, the quality of source texts is increasing, making them more difficult to detect. Identifying student assignments generated by AI presents unique challenges compared to general content. Mainly because of the specificity and contextuality of academic assignments. By adapting modern plagiarism detection methods, the research will contribute to the development of reliable tools that uphold academic integrity and prevent misconduct in student submissions.

In this work we undertake a systematic study to improve the accuracy and reliability of detecting AI-generated text, especially in the educational environment, where plagiarism and academic dishonesty are becoming increasingly relevant due to the use of generative language models. We focused our search on plagiarism detection for text documents and hence excluded papers addressing other tasks. The study aims to adapt modern plagiarism detection methods for reliable classification of AI-generated texts in the context of the Ukrainian language. For this purpose, a new dataset was created based on paraphrased text fragments generated by ChatGPT, also the mT5 model was fine-tuned for text classification.

Methodology of the study is based on general research methods of analysis and synthesis, experimental testing, and quantitative analysis to comprehensively examine and compare the efficacy and performance of different detectors utilized in machine-generated text detection.

2. Detection methods

Currently, numerous detectors have been developed to detect AIGTs. According to the MGTBench [1, p. 3], these detectors are broadly divided into two categories: metric-based and model-based detectors, some of which have shown high accuracy and robustness. While these detectors have been applied in controlled settings, recent studies have explored their effectiveness in real-world scenarios. Metric-based detectors use pre-defined metrics, such as log-likelihood values and rankings, to capture the characteristics of texts and identify AIGTs. In contrast, model-based detectors rely on trained models to distinguish between AIGTs and HWTs [2, p. 2].

Algorithms such as DetectGPT [3], RADAR [4], Ghostbuster [5], GPT-Sentinel [6] amongst others were developed to identify AI-generated content. Approaches to machine text recognition can be divided into four categories [6, p. 2].

Traditional statistical approach of analyzing statistical anomalies in a text sample. By examining statistical differences in language use, such as probability distributions or specific features, zero-shot methods can distinguish human writing from GPT-generated text, leveraging both shallow and deep characteristics. For shallow features, HowkGPT [7, p. 2] computes perplexity scores, establishing thresholds to distinguish their origins. Perplexity measures how well a probability model predicts a sample and is used to compare the performance of different models on the same dataset.

The perplexity of the sequence X can be mathematically expressed through the following function [7, p. 2]:

$$PPL(X) = exp\left\{-\frac{1}{t}\sum_{i}^{t} logp_{\theta}(x_i \mid x_{<i})\right\},$$

where $logp_{\theta}(x_i \mid x_{<i})$ is the log-likelihood of the *i*-th token conditioned on the preceding tokens $x_{<i}$, given θ which represents the parameter values of the model or else the values of the tokens in a given context.

In the context of deep features, DetectGPT [3, p. 3] assumes that machine text always lies in the negative curvature region of the model's log probability function. Based on this hypothesis, DetectGPT transforms the input text using a mask-filling language model, such as T5. It then detects the AI-text by comparing the probabilities of the text and its filled-in variants. Existing zero-shot detectors primarily rely on statistical features, leveraging pre-trained large language models to gather them. These features encompass a range of measures, including relative entropy and perplexity, bag-of-words, average probability, and top-K buckets, likelihood, and probability curvature.

Supervised learning approach of fine-tuning the language model with or without adding a classification module. This approach entails fine-tuning language models on a mixture of human-authored and LLM-generated texts, enabling the implicit capture of textual distinctions.

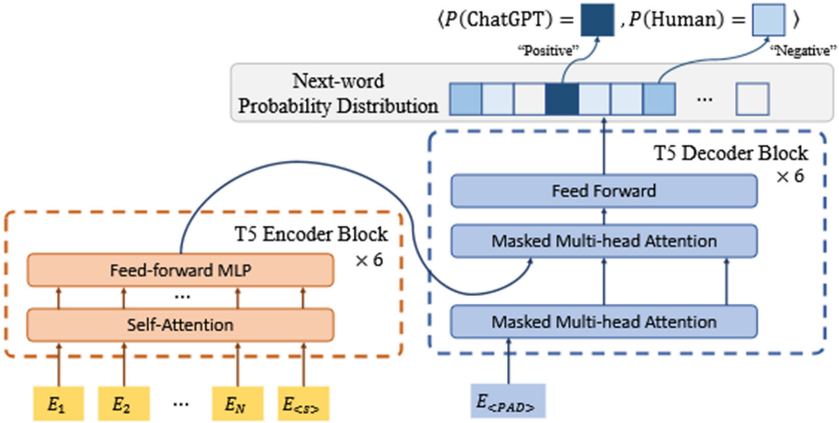

Figure 1. Architecture for T5-Sentinel [6, p. 6]

Despite the strong performance under the supervised learning paradigms, obtaining annotations for detection data can be challenging in real-world applications, leading the supervised paradigms inapplicable in some cases.

While deep learning approaches often yield superior detection outcomes, their black-box nature severely restricts interpretability. Consequently, researchers typically rely on interpretation tools to comprehend the rationale behind the model's decisions [8, p. 5].

Table 1

Key characteristics of detection methods

Method	Approach	Strengths	Weaknesses	Examples
Metric-based	Uses predefined metrics (perplexity, log-likelihood, ranking, etc.)	Fast, interpretable, does not require training	Less effective against advanced evasion techniques	Perplexity-based detection, HowkGPT [7]
Model-based	Uses trained models to classify texts as AIGT or HWT	Higher accuracy, adaptable to new AI models	Requires significant computational resources	DetectGPT, GPTZero [22], Ghostbuster [5]
Supervised learning	Fine-tunes models using labeled datasets of AIGT and HWT	High detection accuracy, learns complex text patterns	Needs labeled datasets, prone to adversarial attacks	OpenAI RoBERTa-based classifier [9], GPT-Sentinel
Unsupervised learning	Learns text patterns without labeled training data	No need for labeled datasets, adaptive learning	Lower accuracy than supervised methods	Clustering-based anomaly detection
Watermarking	Embeds detectable patterns in AIGT to distinguish it	Provides strong ownership tracking, difficult to bypass	Requires AI models to adopt watermarking, paraphrasing reduces effectiveness	Kirchenbauer's watermarking [10], PersonaMark [13]

Watermarking techniques, such as deep learning-based methods, can also be applied to LLMs. The authors of [10] propose a method involving inserting signatures during the decoding stage. These methods categorize the vocabulary into «red» and «green» lists, restricting the LLM to decoding tokens from the green list. Subsequently, The authors of [11] and [12] suggest various algorithms for splitting the red and green lists or sampling tokens from the green list's probabilistic distribution to enhance the interpretability and robustness of watermarking mechanisms during the inference process. PersonaMark [13] is a personalized text watermarking method that leverages sentence structure and user-specific hashing. By embedding unique watermarks, it guarantees copyright protection and user tracking of generated text while maintaining the text's naturalness and generation quality However, the effectiveness of many existing AI-content detectors is significantly reduced due to text paraphrasing.

Two months after OpenAI released ChatGPT, the AIGC detection tool was also introduced. However, OpenAI states that the detector is not fully reliable. Similarly, several AIGC detector tools and software such as CopyLeaks, Turnitin, GPTZero, and Crossplag have been released for the general use of the public to identify AI-generated content. On the other hand, different techniques to attack or evade such AIGC detectors have also been developed and are an active area of research. Evasion techniques such as prompting, recursive paraphrasing, authorship obfuscation, and sentence or word substitution have been developed to point out the failures in the AIGC detector tools.

3. Plagiarism detection

Academic misconduct refers to actions that violate the originality of academic work, such as plagiarism, ghostwriting, data fabrication, any kind of deceit, and content generation using artificial intelligence. Recent generative AI tools are capable of generating various types of content, including text, images, video, and code in multiple programming languages.

The typology of plagiarism varies according to data type or level of obfuscation. The authors of [16, p. 9] presented different typologies defined in several research papers and put forward a new typology for plagiarism according to the level of obfuscation as character-preserving plagiarism, syntax-preserving plagiarism, semantics-preserving plagiarism, idea-preserving plagiarism, and ghostwriting. The plagiarism type may also vary according to the data types, such as code plagiarism and text plagiarism. Plagiarism detection may also be categorized based on factors, such as the number of languages used in a single text; monolingual or cross-lingual detection; extrinsic or intrinsic detection. When plagiarism is detected only using the text itself, it is intrinsic plagiarism detection. Whereas if plagiarism is detected in comparison with other text, it is extrinsic plagiarism detection.

Table 2

Plagiarism detection techniques

Approach	Description	Strengths	Weaknesses
Intrinsic detection	Identifies inconsistencies without external comparison	Detects AI-generated content and ghostwriting	Difficult to detect heavily paraphrased content
Extrinsic detection	Compares text against external sources to find similarities	Detects direct copying and modified text from known sources	Requires an extensive reference database
N-gram-based detection	Uses sequences of n-grams (words or characters) to find similarities	Detects character-preserving and syntax-preserving plagiarism	Difficult to detect heavily paraphrased content
Vector-based detection	Converts text into numerical vectors and measures similarity	Detects idea-preserving plagiarism and AI-generated text	Computationally demanding for large datasets
Syntax-based detection	Analyzes sentence structures and grammar patterns	Detects rewritten but structurally similar content	Less effective for idea-preserving plagiarism
Semantic-based detection	Uses NLP techniques to compare meaning rather than exact words	Detects paraphrased and AI-generated text	Complex and computationally demanding
Fuzzy-based detection	Uses fuzzy logic to identify approximate similarities	Detects minor modifications like spelling variations	Less precise for high-level obfuscation
Stylometric-based detection	Analyzes writing style (sentence length, word usage, etc.) to detect inconsistencies	Detects ghostwriting and AI-generated content	Requires a prior writing sample for comparison

A typical plagiarism detection algorithm involves feature engineering, classification models or text-matching similarity metrics. Plagiarism detection algorithms use common features, such as frequency of characters, average word length, average sentence length, word N-grams frequency, part of speech, synonyms, and hypernyms. Plagiarism is mainly evaluated based on textual similarity with other reference texts. To calculate such similarity, hamming distance, Levenshtein distance, and longest common subsequence distance are the most commonly used string similarity metrics. The most commonly used vector similarity metrics are Jaccard coefficient, Cosine coefficient, Manhattan distance, euclidean distance, Matching coefficient and Dice coefficient [17, p. 5].

Identifying student assignments generated by AI presents unique challenges compared to identifying general AI-generated content. One of the key reasons is the specificity and contextuality of academic assignments. These assignments often require the application of specific theories, principles, and problem-solving skills. While AI-generated general content might exhibit noticeable inconsistencies in the broader context, the narrower and more structured scope of academic assignments may mask such anomalies, making the detection process more complex. Hence, distinguishing AI-generated homework requires more refined and context-aware algorithms.

Applying only technical measures to detect plagiarism will not solve the problem of academic cheating. Educational institutions should consider introducing alternative educational solutions. A possible approach for preventing academic misconduct can be to change current assessment strategies in universities. Educators should place greater emphasis on the student's critical thinking and problem-solving skills rather than just their ability to memorize information. This is particularly important since LLMs can easily provide answers to fact-based questions. With this in mind, teachers should design assignments that encourage creativity and critical thinking, projects that require deep analysis and application of problem-solving skills.

4. Challenges

A detector should reliably distinguish AI-generated texts to ensure that the integrity of content remains intact and to prevent the misuse of LLMs. However, the cost of misidentification by a detector can be significant. If the false positive rate of the detector is too high, students could be falsely accused of AI plagiarism. Existing research has shown that text-matching software not only does not find all plagiarism, but might also incorrectly label highly formal or technical human writing as AI-generated, thus providing false positive results. As a result, the practical applications of AI-text detectors can become unreliable and invalid.

As AI models continue to evolve, the detectors themselves must also adapt to maintain high levels of performance and accuracy. The ongoing race between text generators and detection systems presents a dynamic challenge in the field of content authenticity and security. Additionally, adversarial methods have been developed by AI practitioners to intentionally alter the output of language models to evade detection. These methods can include changes in phrasing, structure, or the introduction of artificial noise that confounds detection systems.

Table 3

Impact of evasion techniques on detection accuracy

Technique	Description	Impact	Affected detectors
Prompting	Crafting AI inputs to generate more human-like responses	Reduces effectiveness of perplexity-based metrics	Metric-based
Paraphrasing	Rewriting text while maintaining the original meaning	Decreases accuracy by altering surface-level features	Metric-based and supervised model-based
Recursive paraphrasing	Iteratively rewording AI-generated text to disguise AI patterns	Greatly reduces detection accuracy by removing statistical patterns	Zero-shot and model-based
Obfuscation	Modifying text to confuse detectors (misspellings, special characters)	Can evade model-based classifiers trained on clean data	Supervised and unsupervised learning models
Substitution	Replacing words with synonyms or altering sentence structures	Reduces reliance on exact-word matching and perplexity analysis	Metric-based and model-based
Authorship obfuscation	Mimicking human stylistic variations to disguise AI-generated text	Makes AI text appear more human-like, reducing model confidence	Supervised learning, watermarking methods

The accuracy and reliability of AI-generated text detection tools can vary depending on several factors, such as the specific tool used, the type of AI model generating the text, and the content being analyzed. Most of the detection tools achieve a 70-80% accuracy rate in detecting text generated by models like GPT-3. Detectors also struggle with short text paragraphs and with more advanced outputs from later-generation models like GPT-4.

In general, detectors have been found to mark HWT as AI-generated (false positives) and AIGT as human-written (false negatives). The detection tools present a bias towards classifying the output as AI-generated rather than human-written.

Detection tools are also shown to be unable to cope with texts translated from other languages. According to a report released by OpenAI, their AI-text detector is not fully reliable on that front. In the reported evaluation of some challenging cases for English texts, their classifier only correctly identifies 26% of AIGT (true positives) while incorrectly classifying 9% of HWT (false positives).

A recent study [18, p. 2] found that state-of-the-art AI-text detectors demonstrated severely degraded performance when encountering texts written by non-native English speakers. It is difficult even for a human to differentiate because of the high capability of LLMs nowadays to produce more and more human-like text. Human essays have more personal experiences, spelling and grammar errors. In contrast, machine essays have more repetitive examples and expressions, according to a quantitative analysis performed by multiple researchers between human and machine-written essays [19, p. 6].

Major differences between HWT and AIGT:

– AI gives organized responses with clear, structured paragraphs or bullet points. The transitions between ideas can be smooth, but more often than not texts seem overly polished or even mechanical. HWT on the other hand tends to have more variability in structure. There might be occasional disorganization, or shifts in tone, that emerge from natural thought flow, making the text feel less «perfect» than AIGT.

– AI tends to use neutral, formal tones unless instructed otherwise. It avoids emotional extremes, unlike humans, who add emotional cues, sarcasm, humor, etc. HWT is often more dynamic, with varied tones depending on the situation. Overall, humans tend to inject more personality into their writing.

– AI is good at recombining existing knowledge, but it's not truly creative in the human sense. Its responses are derivative, drawing from a large database of knowledge, and its examples might not feel fresh or original. Humans excel at creative thinking, often introducing new perspectives, unexpected ideas, and unique problem-solving approaches.

– AI is trained to be contextually aware but has limitations when dealing with highly nuanced or subtle contexts. AI can offer responses that are overly literal, because it relies too much on generalized knowledge rather than specific personal experience.

In long-form responses, AI tends to maintain coherence for a while but can start to lose focus or repeat itself after several paragraphs. Humans are better at maintaining long-term coherence, but they might also repeat ideas or introduce shifts in perspective.

While AI produces grammatically correct and error-free content, it can still make errors in reasoning or factual accuracy that humans might easily catch. It can also occasionally generate awkward phrases, making text seem stiff or unnatural. HWT is more likely to have occasional grammatical errors, typos, missed words, etc.

These summarized features indicate that AI has improved notably in NLP tasks for a wide range of domains. Compared with humans, it may lack individuality but can have a more comprehensive and neutral view towards questions. In short, AIGT tends to be grammatically polished, structurally clear, and neutral in tone, with a tendency toward precision and generalization. On the other hand, HWT tends to have more variability in tone, structure, and depth, with unique personal touches and occasional imperfections that make it seem more authentic.

A modern detector of content generated by LLMs should have the following key characteristics:

– ***accuracy*** means the model should be able to distinguish between LLM-generated and HWT while achieving an appropriate trade-off between precision and recall rates;

– ***data efficiency*** means that the detector should be able to operate with as few examples as possible from the language model;

– ***generalizability*** means that the detector should be able to work consistently, regardless of any change in the model architecture, prompt length, or training dataset;

– ***explainability*** means the detector should provide clear explanations for the reasoning behind its decisions.

Explainability aims to provide human-interpretable reasoning for detection decisions, typically presented as natural language explanations or visual representations of salient features. The task can be further categorized into three levels: ***direct explanation*** (direct identification of forgery clues with few-shot in-context examples), ***reasoning-based explanation*** (multi-hop reasoning and logical consistency evaluation), and ***free-form fine-grained analysis*** (fine-grained analysis of forgery aspects, aligned with a predefined taxonomy of forgery cues). For a given input X, generate an explanation E that:

1) identifies relevant forgery clues $C = \{c_1, c_2, \ldots, c_k\}$;

2) supports multi-layer forgery analysis (low-level, mid-level, high-level).

Traditionally, detecting LLM-generated text is often framed as a binary classification task. However, there is also an «undecided» category, which is used to represent ambiguous texts that may originate from either humans or AI. This category is crucial for enhancing the explainability of detection results. By incorporating it, the system not only improves its reliability but also allows ordinary users to better understand the detection outcomes [20, p. 5].

As AI models continue to evolve, detection tools must improve to handle the increasingly sophisticated outputs the models produce. Until then, educators should approach these tools with caution, recognizing their limitations and the possibility of both false positives and false negatives.

5. Datasets

Many state-of-the-art AI-text detectors show significant performance degradation when faced with texts created by non-native English speakers. Texts by non-native speakers of English are disproportionately flagged as AI-generated [18, p. 6]. So, despite promising results, current models have certain limitations. One of them is that most models are trained only on an English-language corpus of texts. As a result, the efficiency of text recognition in other languages, including Slavic languages, is not optimal. To overcome this limitation, it may be useful to train the models on Ukrainian texts.

To solve this problem, we chose a supervised learning approach to distinguish between HWT and AIGT. First, we collected text fragments for our own datasets and then fine-tuned a language model on two datasets for classification. In our case, we used students' assignments and essays as basis for a dataset of paraphrased text fragments. The HumText dataset consists of different questions and answers on topics from selected courses:

1) Algorithms and data structures;
2) Object oriented programming in C#, Java, Python;
3) Database development;
4) Web development;
5) Operation systems;
6) Software testing;
7) Computer networks.

The GPTText dataset consists of paraphrased text fragments generated by a language model. The dataset contains 26,819 text fragments, each of which corresponds to a text fragment from the HumText dataset. Fragments that were longer than 2,000 words were filtered out. The paraphrasing procedure used the OpenAI API on the gpt-4 model.

Table 4

The most frequent Ukrainian words for humans and AI

Category	**Words**
Humans	наприклад, код, метод, зазвичай, має, проте, алгоритм, водночас, значний, багато, тільки, таким, чином, змінна, клас, отже, зокрема, щоб, незважаючи, замість, крок, відносно, крім, того, проти, дуже, результат, шаблон
AI	це, пояснити, код, алгоритм, можливо, оскільки, задача, функція, суттєвий, наприклад, синтаксис, клас, зазначити, визначити, якщо, існує, ітерація, можливість, завдяки, щоб, так, отже, ідея, залежить, важливо, порівняти, різниця, натомість, чіткий, структура, логіка

We customized the mT5 model [21] for sequence-to-sequence (*seq-to-seq*) classification tasks. The model consists of two main components: an encoder unit and a decoder unit, each of which is repeated 6 times.

The encoder processes the incoming tokens using a self-attention mechanism, after which a multilayer forward propagation is applied. The decoder similarly applies masked multi-head attention and feed-forward layers to the encoded representations, allowing it to generate outputs token by token, predicting the probability of the next word.

The input sequences for the training process consisted of text samples from GPTText, and the output sequences represented the classification result in the form of pos</s> or neg</s>, with </s> as the end-of-sequence marker. The final result is a probability distribution for each following word, which is then used to distinguish between human-generated and ChatGPT-generated sequences.

The fine-tuning process was conducted using Google Colab with TPU acceleration for efficient model training. Fine-tuning was implemented using the Hugging Face Transformers library to ensure compatibility with pre-trained multilingual models. Our model was fine-tuned for 10 epochs. The training setup itself included a batch size of 16 and a learning rate of 3e-5, using the AdamW optimizer. We also employed cross-entropy loss as the objective function to minimize classification errors. Besides that, we used dropout regularization with a probability of 0.1 to enhance generalization.

6. Evaluation metrics

Accuracy measures the proportion of correctly classified instances (true positives and true negatives) out of the total number of instances, which is widely used in classification tasks like multiple-choice task of question-and-answer. The formulation is shown as follows:

$$Accuracy = \frac{True\,positives + True\,negatives}{Total\,samples}$$

F1 score measures the degree of similarity between the labeled and predicted responses obtained from the model. F1 score strikes a balance between precision and recall, offering an all-encompassing assessment of performance, which is especially valuable for binary classification tasks. Precision shows the proportion of true positives among predicted positives while recall among actual positives. F1 score is defined as

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

The metrics like accuracy only consider an average case and are not enough for security analysis. In order to know if the detector can reliably identify the LLM-generated text, researchers need to consider the low false-positive rate regime (FPR) and report a detector’s true-positive rate (TPR) at a low false-positive rate. This objective of designing methods around low false-positive regimes is widely used in the computer security domain.

This is especially crucial for populations who produce unusual text, such as non-native speakers. Such populations might be especially at risk for false-positives, which could lead to serious consequences if these detectors are used in the education system [8, p. 8].

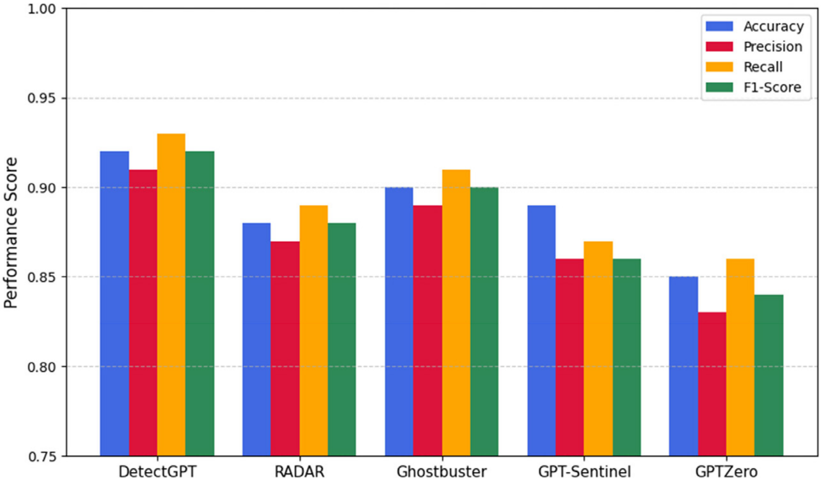

Figure 2. A performance graph of different AI text detectors

The performance of the mT5-Base model was evaluated using three different evaluation metrics: F1, FPR, and FNR (false negative rate). Here, «positive» means that the input text was generated by ChatGPT, while «negative» means that the data was written by a human. Considering the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) results, the metrics are calculated as follows:

$$\mathrm{F1Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN} \qquad TNR = \frac{TN}{TN + FP} \qquad FNR = \frac{FN}{FN + TP}$$

Accuracy, F1, FPR, and FNR for the mT5 model were calculated on the GPTText dataset. The evaluation results are shown in Table 5. All data are presented as percentages.

Table 5

Models accuracy metrics

Model	Accuracy	F1-Score	TPR	TNR
mT5-Base	92.74	93.2	94.76	90.68
DetectGPT [3]	92.03	92.1	94.38	91.04
RADAR [4]	88.15	88.23	89.12	87.45
Ghostbuster [5]	90.24	90.13	91.16	88.42
GPT-Sentinel [6]	89.31	86.84	88.19	86.25
GPT Zero [22]	85.27	84.71	86.09	84.12

The results of the study show that the customized model effectively detects differences between the two types of text, provides some insight into the strengths and weaknesses of the model, and demonstrates its potential for application to practical tasks.

7. Conclusions

In this paper, the differences between text generated by ChatGPT and text written by humans are identified using a language model. For this purpose, we collected a dataset consisting of paraphrased content generated by ChatGPT. After that, the mT5 model was trained to classify the text. This model achieved excellent results, with almost 93% accuracy on a test dataset evaluated using various metrics. Such results provide important information about the effective use of language models for recognizing generated text.

This study will help educational institutions implement more effective plagiarism detection systems that can accurately differentiate between HWT and AIGT in Ukrainian. Our findings can also guide educators in creating fair and transparent regulations for the use of AI-generated content in academic settings.

However, a model trained for a classification task on a dataset like GPTText may not perform well on other NLP tasks for which ChatGPT is widely used, such as answering questions. In the future, we plan to collect datasets with different textual contexts to evaluate the accuracy of the customized mT5 model for different tasks.

The creation of high-quality datasets with AI-generated content are crucial for advancing research in detection and analysis of AI generated content. These datasets are vital for understanding the nuances of AI-generated language, including stylistic patterns, lexical choices, and syntactic structures that distinguish AIGT from HWT. By developing diverse and representative datasets, researchers can ensure that detection systems are not only effective but also adaptable to different domains and languages. Additionally, in-depth analysis of data quality issues will support the creation of high-quality detection models, driving technological advancements and practical adoption in AI-generated media detection.

Future research should explore the application of the mT5 model to other languages and diverse content types. Expanding the dataset to include various styles and contexts will allow for a more robust evaluation of the model's performance. Additionally, investigating models that can detect AI-generated content in more complex settings, such as interactive dialogues or creative writing, could provide a deeper understanding of AI's role in content creation.

References:

1. He, X., Shen, X., Chen, Z.J., Backes, M., & Zhang, Y. (2023). MGTBench: Benchmarking Machine-Generated Text Detection. *Conference on Computer and Communications Security*. URL: https://arxiv.org/pdf/2303.14822
2. Sun, Z., Zhang, Z., Shen, X., Zhang, Z., Liu, Y., Backes, M., Zhang, Y., & He, X. (2024). Are We in the AI-Generated Text World Already? Quantifying and Monitoring AIGT on Social Media. DOI: https://doi.org/10.48550/arXiv.2412.18148
3. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C.D., & Finn, C. (2023). DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. *International Conference on Machine Learning*. DOI: https://doi.org/10.48550/arXiv.2301.11305
4. Hu, X., Chen, P., & Ho, T. (2023). RADAR: Robust AI-Text Detection via Adversarial Learning. DOI: https://doi.org/10.48550/arXiv.2307.03838
5. Verma, V. K., Fleisig, E., Tomlin, N., & Klein, D. (2023). Ghostbuster: Detecting Text Ghostwritten by Large Language Models. *North American Chapter of the Association for Computational Linguistics*. DOI: https://doi.org/10.48550/arXiv.2305.15047
6. Chen, Y., Kang, H., Zhai, V., Li, L., Singh, R., & Ramakrishnan, B. (2023). GPT-Sentinel: Distinguishing Human and ChatGPT Generated Content. DOI: https://doi.org/10.48550/arXiv.2305.07969
7. Vasilatos, C., Alam, M., Rahwan, T., Zaki, Y., & Maniatakos, M. (2023). HowkGPT: Investigating the Detection of ChatGPT-generated University Student Homework through Context-Aware Perplexity Analysis. DOI: https://doi.org/10.48550/arXiv.2305.18226

8. Tang, R., Chuang, Y., & Hu, X. (2023). The Science of Detecting LLM-Generated Text. *Communications of the ACM, 67*, 50-59. DOI: https://doi.org/10.48550/arXiv.2303.07205

9. New AI classifier for indicating AI-written text. URL: https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text, December 2023b. Accessed: 2025-02-27.

10. Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). A Watermark for Large Language Models. *International Conference on Machine Learning*. DOI: https://doi.org/10.48550/arXiv.2301.10226

11. Christ, M., Gunn, S., & Zamir, O. (2023). Undetectable Watermarks for Language Models. *IACR Cryptol. ePrint Arch., 2023*, 763. DOI: https://doi.org/10.48550/arXiv.2306.09194

12. Zhao, X., Ananth, P.V., Li, L., & Wang, Y. (2023). Provable Robust Watermarking for AI-Generated Text. DOI: https://doi.org/10.48550/arXiv.2306.17439

13. Zhang, Y., Lv, P., Liu, Y., Ma, Y., Lu, W., Wang, X., Liu, X., & Liu, J. (2024). PersonaMark: Personalized LLM watermarking for model protection and user attribution. DOI: https://doi.org/10.48550/arXiv.2409.09739

14. Lu, N., Liu, S., He, R., & Tang, K. (2023). Large Language Models can be Guided to Evade AI-Generated Text Detection. *Trans. Mach. Learn. Res*. DOI: https://doi.org/10.48550/arXiv.2305.10847

15. Krishna, K., Song, Y., Karpinska, M., Wieting, J., & Iyyer, M. (2023). Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. DOI: https://doi.org/10.48550/arXiv.2303.13408

16. Foltynek, T., Meuschke, N., & Gipp, B. (2019). Academic plagiarism detection: a systematic literature review. *ACM Computing Surveys (CSUR), 52(6),* pp. 1-42. DOI: https://doi.org/10.1145/3345317

17. Pudasaini, S., Miralles-Pechuán, L., Lillis, D., & Salvador, M. L. (2024). Survey on Plagiarism Detection in Large Language Models: The Impact of ChatGPT and Gemini on Academic Integrity. URL: https://arxiv.org/pdf/2407.13105

18. Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., & Zou, J.Y. (2023). GPT detectors are biased against non-native English writers. *Patterns, 4*. DOI: https://doi.org/10.48550/arXiv.2304.02819

19. Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., & Wu, Y. (2023). How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. DOI: https://doi.org/10.48550/arXiv.2301.07597

20. Zou, Y., Li, P., Li, Z., Huang, H., Cui, X., Liu, X., Zhang, C., & He, R. (2025). Survey on AI-Generated Media Detection: From Non-MLLM to MLLM. DOI: https://doi.org/10.48550/arXiv.2502.05240

21. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2020). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. *North American Chapter of the Association for Computational Linguistics*. DOI: https://doi.org/10.18653/v1/2021.naacl-main.41

22. AI Detector – the Original AI Checker for ChatGPT & More. (n.d.). URL: https://gptzero.me/