

# Analyzing the nature of AI footprint: noise-to-text method

Emil Faure<sup>†</sup> and Andrii Nykonenko<sup>\*†</sup>

*Cherkasy State Technological University, 18006 Cherkasy, Ukraine*

## Abstract

In this paper, we discuss the topic of AI writing detection for text data. The real arms race is happening in the AI market these days with three major players: LLM creators, AI Writing detector developers, and actors who are trying to hide or mask the AI footprint. In these circumstances, it is vital to understand better the character of the AI footprint and its main properties. The main goal of this paper is to explore the nature of the AI footprint and find the role it plays in the AI writing detection task. To reach this goal, we propose a new research approach based on a noise-to-text data transformation method. The method offers consecutive data conversion through multiple steps, allowing us to receive a sequence of data versions from completely senseless up to normal text. Each received sample of data, we analyze with a set of AI Writing Detectors to get AI scores. In our research we show that the AI score is more connected to detecting specific writing and thinking styles typical of AI than to the actual amount of AI-generated data in a sample. AI-generated data could be unseen by detectors if it isn't organized into well-known patterns of expressing thoughts and thinking. We hope that the proposed research brings some light into the question of AI footprint nature as well as supports a better understanding of the guiding principles of AI writing detection systems. Understanding the character of the AI footprint is key to building reliable and interpretable AI Detectors.

## Keywords

AI footprint, LLM generated text, AI writing detection1

## 1. Introduction

Recent advancements in creating Large Language Models (LLM) and their new capabilities could not be imagined just a few years ago. The whole scientific and business society has come a long way from the GPT-1 release in June 2018, to the groundbreaking ChatGPT release in November 2022. ChatGPT release was probably one of the biggest factors in the current pace of the rise of new LLMs from different vendors, as it showed how AI could impact the world today and how suitable and helpful AI technologies could be in day-to-day life. Later in 2023 and during 2024 we saw a big number of releases from Google, OpenAI, Meta, Anthropic, and other players who want to reserve a share of the quickly rising AI market.

The approach of releasing LLMs in a chatbot manner significantly changed the way people can collaborate with AI. Before ChatGPT, the only way of interacting with AI for a person without specific Data Science knowledge was to use a specific software that leverages AI. In that scenario the general public was interacting not with a model itself, but with a product where the model plays some role. With the ChatGPT release it was changed, and now a broad audience can interact with a model directly, although within the limits established by the model developer.

The wide adoption of AI created some problems with misuse. As for any other new technology, not all the potential use cases are meant for the greater good, things like fake news, misinformation creation, and academic misconduct are among bad scenarios. To deal with that side of things we

---

*CIAW-2024: Computational Intelligence Application Workshop, October 10-12, 2024, Lviv, Ukraine*

\* Corresponding author.

<sup>†</sup> These authors contributed equally.

✉ e.faure@chdtu.edu.ua (E. Faure); andrey.nykonenko@gmail.com (A. Nykonenko)

ORCID 0000-0002-2046-481X (E. Faure); 0000-0002-9442-1601 (A. Nykonenko)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

need to control the spread of AI-generated information and the AI writing detection software is the only reliable way of achieving that today. In [1] and [2] authors propose the following approaches for AI-generated text identification:

1. Feature-Based Approaches
2. Neural Language Model Approaches
3. Feature-Based + Neural Language Model
4. Watermarking
5. Human-aided
6. Information retrieval-based

The last approach has been proposed in article [2], this method is based on the idea of storing all the AI-generated texts in one database, so the detection task comes down to a search. Approach #5 relies on human-machine collaboration, where additional tools like [3] were used to support human judgment. The other four approaches one way or another rely on token distribution generated by a model. There are a lot of practical recommendations in the literature, for example in [4] on how to properly train an AI writing detector software following proposed principles. The only caveat regarding those first four approaches is that we rely on one black box to identify the results of the other black box. With feature-based approach and watermarking methods, this could be a bit less a case as we at least have a basic understanding of the nature of underlying features, but for both approaches #2 and #3 this is the reality.

Even if the existing generation of AI writing detectors based on the mentioned approaches works pretty well [5] it still includes significant risks related to the absence of understanding key features and foundational principles lay under the AI detection technology. The absence of this knowledge puts into question the long-term reliability of AI writing detection technology. Also, we can't make an educated guess about the theoretical boundaries of the use of the current technology and what are normally supported cases. This understanding is crucial for new technology trustworthiness and can't be received without understanding the foundational principles lying behind text generation and AI detection.

One of the most active battlefields in this situation is the education market. All players invest in research related to understanding the principles used by an LLM for text understanding and generation to get some advantages in the arms race. Despite the fact that all parties achieved significant progress in their field there is still a lack of deep understanding of basic principles that allows LLM to achieve such a high proficiency in text-related tasks. To the best of our knowledge, recent attempts to use interpretability and explainability techniques for LLMs showed very limited effectiveness. We assume that understanding the nature of AI footprint on text data and its differences from human footprint is key to a deeper understanding of both data generation principles, AIW detection, and AI footprint hiding techniques. The current article proposes and discusses one approach for analyzing the AI footprint nature focused on AIW detection capabilities.

## 2. Related work

Area of the AIW Detection is a hot research topic these days. Multiple research papers are dedicated to different approaches that could be used for AI Detection and comparison and quality analysis of available detectors. In the previous section, we already mentioned the main state-of-the-art approaches that currently dominate the scientific and commercial AI communities.

The detector's quality analysis is one of the key topics tightly connected to AI detection software creation approaches and architecture choices. There are multiple domain-specific studies:

- AI detection in the medical domain [6], authors use AI-generated rehabilitation-related articles to compare the accuracy of mainstream AI writing detectors and human reviewers; also they analyze the influence of paraphrasing on AI detector's ability to find AI content.

- Short-form physics essay submissions analysis performed in [7], where authors mention that AI detectors perform significantly better compared to human judges for distinguishing AI and human authorship.
- Article [8] discusses challenges related to AI usage in academic writing and the ethical boundaries and acceptable scenarios.
- In work [9] authors discuss the use of AI for solving coding problems and corresponding AI detector performance for distinguishing between human-written and AI-generated Python solution codes.

General AI Detectors comparison is also a quite well-developed topic, multiple works show that in general AI detectors propose a very good opportunity to differentiate between human-written and AI-generated data. For example, a study [5] shows a result of a massive comparison of 16 AI detectors. The author evaluated the performance with different metrics: overall accuracy, accuracy per document type, the number of uncertain responses, the number of false positives, and the number of false negatives. The study shows that at least three detectors have very high accuracy across all proposed documents and most of the other detectors can distinguish between GPT-3.5 papers and human-generated papers with reasonably high accuracy. Another paper [10] examines the general functionality of detection tools for artificial intelligence-generated text and evaluates them based on accuracy and error type analysis. It covers 14 AI detection tools on 6 different datasets. Received results show that most of the analyzed detectors are quite good on non-modified data (including machine-translated data), but not as stable in case of applying additional content obfuscation techniques. The authors of the [11] survey aim to provide a concise categorization and overview of current work encompassing both the prospects and the limitations of AI-generated text detection. They analyze different text modification attacks on detectors and the potential social consequences of the impossibilities of AI-generated Text Detection.

Based on analyzed sources we can conclude that in general AI detection tools work quite well for scenarios they are intended to work on, specifically distinguishing between human-written and AI-generated text. Some specific limitations should be mentioned to be sure that a tool works in the intended scenario:

- Analyzed text hasn't been modified
- AI data is produced by LLM that is supported by the tool
- The text contains long spans of the same nature

Multiple research papers say that AI Writing detection tools are pretty efficient in the identification of data generated by LLMs [12], [13], [14]. The situation with falsely marking human-written text as AI-generated varies from detector to detector, but seems to be improving over time [10]. Some AI detectors have more limitations and some are more stable, support data generated by a wide range of LLMs, or produce consistent output despite adversarial prompting or other text manipulation techniques. In general, the AI Detection area is a pretty quickly evolving field and there is a time gap between the emergence of a new LLM or a text modification technique and the time when it will be added to detectors.

### 3. Method

During our research, we have observed some interesting common patterns that stay consistent across multiple AIW detection systems. Based on those observations we assume that most of the existing AIW detection tools don't catch the whole LLM footprint. Most probably they can catch a partial footprint or even some specific artifacts only. To bring some light onto this topic and to improve general understanding of basic principles of AIW detection we propose to make a series of experiments to support or disprove this idea. We want to define a sharp edge between LLM-generated data that could be detected and that couldn't. We are talking about defining a detection

sensitivity threshold. LLM-generated text that stays lower than the threshold most probably wouldn't be detected as AI-written and that which stays above the threshold would be detected. We hope that introducing the threshold idea opens a door for deeper research and understanding of key differences in the nature of LLM-generated data between both sides of the threshold. Knowing those key differences we will be able to better understand the nature of the separating hyperplane for AIW detectors and what task they are trying to solve.

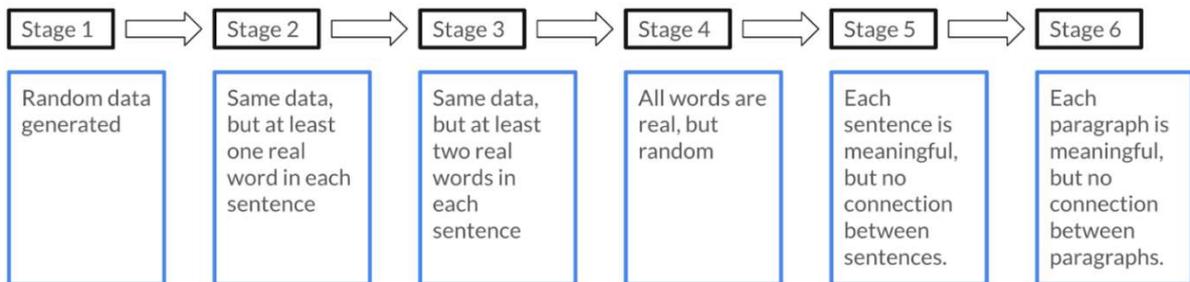
As a basic method for this research, we will be using noise to text approach proposed in this study. In this approach, we will use an iterative process for generating text data using LLM. The process starts with a complete noise generation and continues by iteratively increasing generated data quality. All the data quality improvement stages as well as original noise generation are made by LLM. We expect that data generated in the first stage is a completely senseless but unique combination of characters while data received in the last stage is a good quality, meaningful, and readable text. All the changes should be made by LLM only to keep the LLM footprint clear and not to introduce any human or other tool footprint by accident.

We will be measuring the AIW score for each stage using a combination of AIW Detection tools. If our idea is right we expect to see an average AIW score very low for the first few stages with an increase for the middle stages and close to 100% for the latter stages. The expected switch from low scores to high scores should indicate the existence of the detection sensitivity threshold and the potential space where it lies.

## 4. Data

To generate data according to the proposed noise-to-text method we define 6 main stages (Figure 1) that will be responsible for sequential transformation of noise. All the transformations are made by an LLM to reduce the probability of the introduction of any side interference on the footprint. All data were generated by gpt-3.5-turbo-0125 [15] which is the latest available model among the GPT-3.5 family at the moment of the research (May 2024). For the sake of reproducibility of experiments, we were trying to store initial prompts for each stage. At the very first stage, we generated 10 texts that were passed through all other stages. For each following stage, we used text from the previous stage as input. At the end of the experiment, we calculated the AIW score for each text on each stage.

Based on our experience it was almost impossible to generate the required response with just one prompt, especially for later stages. We assume that this effect is related to the influence of the text proposed for modification, as each prompt for each stage except the first one was constructed from two parts: instructions to LLM describing the nature of the requested change and the text to modify regarding proposed instructions. In some cases having the same instructions but different text as input behavior of the LLM was very different, varying from exactly following instructions up to completely refusing to execute the request. We have applied a few different techniques to deal with this behavior inconsistency. Most of the time slight changes in the original instructions formulation helped, in other cases only a long dialog with the LLM or splitting the original instruction into a set of steps helped. For reproducibility purposes we stored all the variants of initial instructions applied at each stage, they can be obtained there [16].



**Figure 1:** Main transformation stages applied to a text.

- Stage 1.** This stage is responsible for initial noise generation. In the generated data we were trying to preserve normal text structure, to avoid any structural or formatting bias. That's why we asked the LLM to generate a noise that consists of "words", combined into "sentences" and "sentences" combined into "paragraphs". In some cases, the LLM refused to follow instructions, so the data we use as a final output of the stage is the result of picking the best sample from generated data after multiple attempts. In general, it was surprisingly hard to get a unique senseless text from the LLM. The hardest part for the LLM was the concept of unique senseless words, in some cases, it understands that as "fake words", relates it with fake data generation, and refuses to follow other instructions because of the model's rules and policies. In other cases, it started to use the typical Lorem ipsum lexic [17] using the whole preexistent phrases or even sentences. As we are interested in unique LLM-generated noise we used Google search to avoid mistakenly adding Lorem ipsum-based texts to our collection. In other cases, it understands the instruction as a prohibition to use only English words and generated data in Spanish or German. Probably across all the stages, this one took the most time and effort.
- Stage 2.** In this stage, we make the first step from text that consists of totally random characters towards the first meaningful components. The original idea was to ask LLM to change every fifth word to a real word at this stage, then every fourth, third, second, and all the words in the following stages to receive a smoother transition from random noise to real text. Experiments show that in general LLM doesn't understand such difficult concepts as "every fifth word", in some cases it's able to more or less reliably execute tasks related to "every second word", but nothing more than that. Even splitting the original task into multiple simple subtasks with the introduction of indexes and asking to replace each word with an index in a certain list didn't work well. After a series of experiments, we figured out that the only way to coerce a model to execute such a task is one way or another to program it on a very detailed level, but there isn't any guarantee that such a program could preserve the original model's footprint. That's why we continued with a simplified, but more natural LLM way of increasing the meaningfulness of a text. Following this principle, task formulation for stage 2 was to replace only the first word in each sentence while preserving all other words, sentences, and paragraph structure as it was in stage 1. In general, the LLM understands the meaning of the concept of the "first word of each sentence" pretty well, so this allows us to receive an output with expected changes in most cases. But at this stage, we faced another problem, which was shown to be common for a few following stages as well. We discovered that in most cases the LLM identifies text generated on stage 1 as fake text and refuses to make any changes to it as a model concerns these changes as spread of misinformation. LLM misinformation policies make this research really hard, but with additional efforts, we were able to receive the needed output for all 10 texts generated in stage 1.
- Stage 3.** This stage is intended to be the second step towards receiving more meaningful content, but still not too meaningful. Following principles, restrictions, and findings discovered earlier we formulated this task as "replace two words in each sentence". Even if in general LLM understands the concept of "two words" pretty well, this turned out to be a very challenging task. After some extensive prompting, we ended up with content that in general satisfies the principles of the proposed method, but some individual sentences can have slightly more or less than two changed words.
- Stage 4.** The expected result of this stage is to have text with the same number of words, sentences, and paragraphs but all words to be normal English words. This task was more or less straightforward for the LLM.
- Stage 5.** If in the previous stage, all words were from a normal English dictionary, but sentences were meaningless, here we construct meaningful sentences. We preserve no connection between individual sentences to keep paragraphs meaningless. For data

generated on this and the next stages it was hard to keep the LLM from generating a new holistic meaningful story, so we used additional tricks to preserve the original text structure in terms of sentences and paragraphs. To do that we asked the LLM to preserve the first word in each sentence, it generally works quite well with some rare exceptions.

- **Stage 6.** For this stage, the main goal is to receive a normal meaningful story in each paragraph, but avoid making a relation between paragraphs to preserve the whole text in a disorderly state. During the experimentation planning phase, we were supposed to have one additional stage after this one. However further experiments showed that even content generated on stage 6 becomes 100% detectable to most detectors, so we skipped previously planned stage 7 as redundant. Another important note about data generation in the current stage is that in the proposed setup the LLM doesn't want to obey any rules regarding text structure, it tends to generate completely new text. We were trying multiple ways to preserve the same amount of sentences in each paragraph as it was in stage 5, but weren't able to find a stable way to do so. Finally, we created a prompt that asks the LLM to generate at least 5 sentences in each paragraph. That doesn't guarantee that LMM would follow this instruction precisely, but that way we were able to receive enough content per paragraph.

As a result of the data generation process at stage 1, we received 10 texts that were written using a combination of random characters instead of words. Then all these texts were passed through stages 2-6, so we received an additional 5 versions of each text, where each stage increased the meaningfulness of the original data. Now we have 10 sets of data that are created following the noise-to-text approach. Examples of prompts used as a starting point for data generation on each stage are available here [16], all the generated data for each stage can be obtained from [18].

## 5. Detectors

There are a lot of available AIW detectors on the market. All of them could be split into two big categories:

1. Tools for checking the authenticity of writing, are mostly used in education and other fields where checking for AI is a part of a standard pipeline for ensuring work originality.
2. Tools that help AI users avoid being caught on their misuse of AI and hide the AI footprint. Typically those tools contain some text modification capabilities (e.g. paraphrasing or other AI detection bypassing techniques). AIW detection capability plays the role of the final assessor, to ensure that the final product of applied techniques is a content undetectable by other AIW detectors.

These two groups of players stand from different sides of the market and actually, there is an ongoing arms race among them. Because of the different purposes, these tools have their specifics which we aren't going to discuss in the boundaries of the following research, but we want to highlight one key difference that could be important for us. This difference is the sensitivity of the tools. In general, talking about the efficiency of AIW detectors we can think of precision/recall or FPR/recall metrics. The first group of tools mainly focused on evaluating students' writing, so FPR became one of the most important measurements for them because the consequences of wrongly marking student text as AI written are big for both the student and the tool owner. Knowing about the precision/recall tradeoff it is easy to conclude, that on average sensitivity of detectors from that group would be lower.

On the other hand, tools from the second group are mostly focused on high recall. In their case, the price of false positives isn't so high compared to false negatives. It wouldn't be a problem if they falsely mark human-written text as AI-generated. But it would be a problem if they did not flag weak paraphrased AI text and got their clients in trouble by being caught by a tool from the first group. That's why it is expected to see on average higher sensitivity of detectors from that group.

We want detectors from both groups to participate in this research. Another important question is the availability of the detectors (some could be free, some available by subscription or price per amount of words, etc.). Among other characteristics important for detectors to be included we took into account:

1. General popularity and rating
2. Quality of the detector
3. Minimal and maximal amount of words able to process
4. Ability to process even senseless text
5. Detection score as a number

Regarding the last item, for the sake of research, we need to get a number that represents the AIW score for the text. Even if a detector also has some additional capabilities (e.g. paraphrased text detection), we still use only AIW score. If a detector returns a human-written score instead of an AIW score we reverse that value using the  $1 - \text{human\_score}$  formula. Most of the detectors can produce an AIW score number but with a different meaning. Some of the detectors return % of text that was AI generated (fraction of AI-generated sentences) while others return % as a probability that the whole text was AI-generated. Even if these two numbers represent different things, when applied to fully AI-generated content they represent the same characteristics of the nature of the text, so we will not make a difference among them.

We have excluded from the consideration all the detectors that can produce only binary (human, AI), ternary, or quintet (human, mostly human, mixed, mostly AI, AI) output without an exact score or percentage. Also, some detectors weren't able to analyze data generated in this research, they declined proposed texts because of wrong language defined or other technical limitations. Another common type of limitation is the max text length limit, some detectors calculate that in words, others in characters; for detectors with such a limit, we cut the ending part of a text to follow length requirements. In case no limitation is applied we use the whole text as presented in [18].

Following the described principles, we ended up using five detectors for research purposes. Three of them are from the first group and two from the second. Selected detectors are shown in Figure 2.



**Figure 2:** Set of detectors used in the research. Left column - authenticity tools; right column – tools could be used for masking AI footprint.

## 6. Results

All the created data were passed through 5 selected detectors, detailed results are available here [18]. Here in Table 1, we present only the final aggregated statistics per stage per detector.

**Table 1**

Average AIW score for 10 texts per detector per stage.

	Turnitin	GPTZero	Originality	Scribbr	Quillbot	Stage average
Stage 1	0%	4%	3%	5%	0%	2.4%
Stage 2	0%	4%	7%	13%	0%	4.8%
Stage 3	0%	5%	11%	8%	1%	5.0%
Stage 4	0%	6%	19%	7%	5%	7.4%
Stage 5	0%	50%	99%	22%	29%	40.0%
Stage 6	75%	87%	100%	87%	81%	86.0%

Talking about the received results we can mention a few important observations:

1. As expected, different detectors have different sensitivity, but what is surprising is that Originality, which belongs to the first group of detectors (authenticity tools) has the highest sensitivity among all of them.
2. In general, for all the detectors except Scribbr, the score does not decrease at each subsequent stage. This supports our idea that the meaningfulness of AI-generated text correlates with the detected AIW score.
3. Received numbers support the original hypothesis that AIW detectors can catch not an LLM footprint, but rather a way of expressing thoughts and thinking typical for an LLM. That's why data from the first four stages received very low AIW scores, even though it was fully AI-generated.
4. It is worth mentioning the significant increase in AIW score between stages 4-5 and stages 5-6. Stage 4 presents a text constructed from normal English vocabulary but organized in a pointless manner. Stage 5 presents comprehended but not related sentences. So the main difference between these two stages is added meaning or sense. The same applies to a distinction between stages 5 and 6, where the main difference is an expansion of meaning from sentence-only level to paragraph level.

## 7. Discussion

In this article, we proposed to discover the nature of the AI footprint using the noise-to-text method. To do that we build the 6 transformation stages process that allow us to receive an AI-generated text with different levels of meaningfulness. Following experiments using 5 different AIW detectors showed that the AI score returned by the detectors correlates with text meaningfulness. The other important observation is that a text generated in the early stages is in general indistinguishable from human writing for these detectors. This implies that the detectors are more focused on the way thoughts are expressed and probably how thinking flows in general than on specific character or word N-gram features. Some additional experiments would be needed to make a stronger foundation for this assumption, but the described research proposes a good starting point for follow-up AI footprint exploration.

We made an extensive analysis of the literature to compare the received results with the works of other authors. Even though AI generation and detection are pretty active research topics these days the main focus stays on just three areas:

- LLMs-related research, mostly centered around the comparison of the different architectures and generated content quality assessment
- General comparison of the AI detector's quality, as was shown in [5], [12], and [11]

- Deeper research and evaluation of the AI detector's performance in some specific cases e.g. modified text [10] or text written by nonnative speakers [19]

Regarding the AI footprint (as we understand it in the current work) most of the mentions are located in marketing articles and AI-masking tools websites e.g. [20] and [21] which is not helpful in the context of the research. Probably, the closest article so far is [3] which describes the GLTR tool and says that the tool "enables forensic inspection of the visual footprint of a language model on input text to detect whether a text could be real or fake". Indeed, GLTR is a pretty helpful tool and it played a crucial role in the pre-AI detection era when no detectors were available on the market. The tool works on a word level and highlights words that are more commonly used in AI writing. Nowadays, it's a bit outdated as it preserves tokens' statistical characteristics inherited from GPT-2 which are quite different from the modern LLMs. Despite this use of GLTR still could be an interesting exercise, especially for analyzing data from earlier generations of LLMs.

Our research supports the conclusion that AIW detectors can detect the AI Writing not in all the cases, but only in the case of at least a few meaningful sentences combined into one passage. To the best of our knowledge, this is one of the first studies that covers a text meaningfulness topic and its impact on AI detection. It's too early to make an exact conclusion about whether it's because meaningless text doesn't have an AI footprint or if there is a significant difference between the footprint of meaningful and meaningless data. Another possible option is that AIW detectors identify not the AI footprint itself, but some other characteristics or artifacts typical for meaningful AI-generated text. We are leaning towards the latter option, but there should be some additional research on the topic.

## 8. Conclusion and further work

This paper describes a new method for defining what AI footprint is and how it could be potentially discovered and later measured. This study doesn't answer the question of what AI footprint is but moves us one step closer to understanding its nature and what other experiments could be conducted to finally define the term. Based on the conducted research we were able to trace a relationship between AI Detectors' ability to recognize generated text and its level of meaningfulness and consistency. We can't affirm unequivocally that found relation is definitively a sign of the AI footprint. Here we rely on a hypothesis that the AI detectors work based on AI footprint, but there is no evidence that this is true. Additional research is needed to support or decline this hypothesis and we will address this in our next studies.

A thing we can state for sure is that the absence of evidence in the AI Detection area leads to a lot of confusion and limits potential use cases for this software. Even assuming the limited performance of the AI Detectors in some specific cases (like the appearance of a new LLM or applying text modification techniques) they could be significantly more helpful if they can provide not just a judgment, but evidence as well. In that case, a human could be the one who makes the final decision – the same scenario exists in plagiarism detection, where software is responsible for finding matches, and a human is responsible for classifying them as a paraphrase or not. We assume that further research dedicated to the AI footprint topic is the essential piece that needs to be solved to allow future progress regarding the AI evidence. We see multiple areas worth researching with regard to their influence on the footprint: paraphrasing, machine translation, AI bypassing techniques, comparison of different LLMs footprint, mixed data, and other text generation/modification techniques. We will be working to cover more of them in the future.

## References

- [1] E. N. Crothers, N. Japkowicz, H. L. Viktor, Machine-generated Text: A Comprehensive Survey of Threat Models and Detection Methods, *IEEE Access* 11 (2023) 70977-71002. doi:10.1109/access.2023.3294090.

- [2] K. Krishna, Y. Song, M. Karpinska, J. Wieting, M. Iyyer, Paraphrasing Evades Detectors of AI-generated Text, but Retrieval is an Effective Defense, *Advances in Neural Information Processing Systems* 36 (2024).
- [3] S. Gehrmann, H. Strobelt, A. M. Rush, GLTR: Statistical Detection and Visualization of Generated Text, arXiv preprint arXiv:1906.04043 (2019).
- [4] D. Yan, M. Fauss, J. Hao, W. Cui, Detection of AI-generated Essays in Writing Assessments, *Psychological Test and Assessment Modeling* 65(1) (2023) 125-144.
- [5] W. H. Walters, The Effectiveness of Software Designed to Detect AI-Generated Writing: A Comparison of 16 AI Text Detectors, *Open Inf. Sci.* 7(1) (2023). doi:10.1515/opis-2022-0158.
- [6] J. Q. J. Liu, K. T. K. Hui, F. Al Zoubi, Z. Z. X. Zhou, D. Samartzis, C. C. H. Yu, J. R. Chang, A. Y. L. Wong, The Great Detectives: Humans versus AI Detectors in Catching Large Language Model-Generated Medical Writing, *Int. J. Educ. Integr.* 20(1) (2024). doi:10.1007/s40979-024-00155-6.
- [7] W. Yeadon, E. Agra, O.-o. A. Inyang, P. Mackay, A. Mizouri, Evaluating AI and Human Authorship Quality in Academic Writing through Physics Essays, *Eur. J. Phys.* (2024). doi:10.1088/1361-6404/ad669d.
- [8] J. Homolak, Exploring the Adoption of ChatGPT in Academic Publishing: Insights and Lessons for Scientific Writing, *Croat. Med. J.* 64(3) (2023) 205–207. doi:10.3325/cmj.2023.64.205.
- [9] W. H. Pan, M. J. Chok, J. L. S. Wong, Y. X. Shin, Y. S. Poon, Z. Yang, C. Y. Chong, D. Lo, M. K. Lim, Assessing AI Detectors in Identifying AI-Generated Code: Implications for Education, in: *ICSE-SEET '24: 46th International Conference on Software Engineering: Software Engineering Education and Training*, ACM, New York, NY, USA, 2024. doi:10.1145/3639474.3640068.
- [10] D. Weber-Wulff, A. Anohina-Naumeca, S. Bjelobaba, T. Foltýnek, J. G. Guerrero-Dib, O. Popoola, P. Sigut, L. Waddington, Testing of Detection Tools for AI-generated Text, *International Journal for Educational Integrity* 19(1) (2023) 26.
- [11] S. S. Ghosal, S. Chakraborty, J. Geiping, F. Huang, D. Manocha, A. S. Bedi, Towards Possibilities & Impossibilities of AI-generated Text Detection: A Survey, arXiv preprint arXiv:2310.15264 (2023).
- [12] A. M. Elkhatat, K. Elsaid, S. Almeer, Evaluating the Efficacy of AI Content Detection Tools in Differentiating Between Human and AI-generated Text, *Int. J. Educ. Integr.* 19(1) (2023). doi:10.1007/s40979-023-00140-5.
- [13] A. Foster, Can GPT-4 Fool TurnItIn? Testing the Limits of AI Detection with Prompt Engineering, [https://digital.kenyon.edu/cgi/viewcontent.cgi?article=1041&context=dh\\_iphs\\_ai](https://digital.kenyon.edu/cgi/viewcontent.cgi?article=1041&context=dh_iphs_ai) (2023).
- [14] A. Akram, An Empirical Study of AI Generated Text Detection Tools, arXiv preprint arXiv:2310.01423 (2023).
- [15] OpenAI, Models Overview, <https://platform.openai.com/docs/models/gpt-3-5-turbo> (2024)
- [16] A. Nykonenko, Analyzing the Nature of AI Footprint: Noise-to-text Method. *Text Generation Instructions*, <https://docs.google.com/spreadsheets/d/1yTDwvn4weVrOTPMmqKpzUGdo2CuuOHHNxzV4mUHiBjc/edit?usp=sharing> (2024).
- [17] Wikipedia, Lorem Ipsum, [https://en.wikipedia.org/wiki/Lorem\\_ipsum](https://en.wikipedia.org/wiki/Lorem_ipsum) (2024).
- [18] A. Nykonenko, Analyzing the Nature of AI Footprint: Noise-to-text Method. *Data and Evaluation*, <https://docs.google.com/spreadsheets/d/1R7yVcS-L-i4usMTkFaO9KbxvR3o7EWVQN8zxV45NSGk/edit?usp=sharing> (2024).
- [19] W. Liang, M. Yuksekogonul, Y. Mao, E. Wu, J. Zou, GPT Detectors Are Biased Against Non-native English Writers, *Patterns* 4(7) (2023) 100779. doi:10.1016/j.patter.2023.100779.
- [20] C. Adam, How to Avoid AI Writing Detection and Safeguard Your Content from Flagging, <https://medium.com/@catalinadam76/how-to-avoid-ai-writing-detection-and-safeguard-your-content-from-flagging-8eccaac2c8ef> (2024).
- [21] UNDETECTABLE LLC, Advanced AI Detector and Humanizer, <https://undetactable.ai> (2024).