International Journal for Educational Integrity

ORIGINAL ARTICLE Open Access

Testing of detection tools for AI-generated text

Debora Weber-Wulff[1], Alla Anohina-Naumeca[2], Sonja Bjelobaba[3*], Tomáš Foltýnek[4], Jean Guerrero-Dib[5], Olumide Popoola[6], Petr Šigut[4] and Lorna Waddington[7]

*Correspondence:
sonja.bjelobaba@crb.uu.se

[1] University of Applied Sciences HTW, Berlin, Germany
[2] Riga Technical University, Rīga, Latvia
[3] Uppsala University, Uppsala, Sweden
[4] Masaryk University, Brno, Czechia
[5] Universidad de Monterrey, San Pedro Garza García, Mexico
[6] Queen Mary University of London, London, UK
[7] University of Leeds, Leeds, UK

Abstract

Recent advances in generative pre-trained transformer large language models have emphasised the potential risks of unfair use of artificial intelligence (AI) generated content in an academic environment and intensified efforts in searching for solutions to detect such content. The paper examines the general functionality of detection tools for AI-generated text and evaluates them based on accuracy and error type analysis. Specifically, the study seeks to answer research questions about whether existing detection tools can reliably differentiate between human-written text and ChatGPT-generated text, and whether machine translation and content obfuscation techniques affect the detection of AI-generated text. The research covers 12 publicly available tools and two commercial systems (Turnitin and PlagiarismCheck) that are widely used in the academic setting. The researchers conclude that the available detection tools are neither accurate nor reliable and have a main bias towards classifying the output as human-written rather than detecting AI-generated text. Furthermore, content obfuscation techniques significantly worsen the performance of tools. The study makes several significant contributions. First, it summarises up-to-date similar scientific and non-scientific efforts in the field. Second, it presents the result of one of the most comprehensive tests conducted so far, based on a rigorous research methodology, an original document set, and a broad coverage of tools. Third, it discusses the implications and drawbacks of using detection tools for AI-generated text in academic settings.

Keywords: Artificial intelligence, Generative pre-trained transformers, Machine-generated text, Detection of AI-generated text, Academic integrity, ChatGPT, AI detectors

Introduction

Higher education institutions (HEIs) play a fundamental role in society. They shape the next generation of professionals through education and skill development, simultaneously providing hubs for research, innovation, collaboration with business, and civic

engagement. It is also in higher education that students form and further develop their personal and professional ethics and values. Hence, it is crucial to uphold the integrity of the assessments and diplomas provided in tertiary education.

The introduction of unauthorised content generation—"the production of academic work, in whole or part, for academic credit, progression or award, whether or not a payment or other favour is involved, using unapproved or undeclared human or technological assistance" (Foltýnek et al. 2023)—into higher education contexts poses potential threats to academic integrity. Academic integrity is understood as "compliance with ethical and professional principles, standards and practices by individuals or institutions in education, research and scholarship" (Tauginienė et al. 2018).

Recent advancements in artificial intelligence (AI), particularly in the area of the generative pre-trained transformer (GPT) large language models (LLM), have led to a range of publicly available online text generation tools. As these models are trained on human-written texts, the content generated by these tools can be quite difficult to distinguish from human-written content. They can thus be used to complete assessment tasks at HEIs.

Despite the fact that unauthorised content generation created by humans, such as contract cheating (Clarke & Lancaster 2006), has been a well-researched form of student cheating for almost two decades now, HEIs were not prepared for such radical improvements in automated tools that make unauthorised content generation so easily accessible for students and researchers. The availability of tools based on GPT-3 and newer LLMs, ChatGPT (OpenAI 2023a, b) in particular, as well as other types of AI-based tools such as machine translation tools or image generators, have raised many concerns about how to make sure that no academic performance deception attempts have been made. The availability of ChatGPT has forced HEIs into action.

Unlike contract cheating, the use of AI tools is not automatically unethical. On the contrary, as AI will permeate society and most professions in the near future, there is a need to discuss with students the benefits and limitations of AI tools, provide them with opportunities to expand their knowledge of such tools, and teach them how to use AI ethically and transparently.

Nonetheless, some educational institutions have directly prohibited the use of ChatGPT (Johnson 2023), and others have even blocked access from their university networks (Elsen-Rooney 2023), although this is just a symbolic measure with virtual private networks quite prevalent. Some conferences have explicitly prohibited AI-generated content in conference submissions, including machine-learning conferences (ICML 2023). More recently, Italy became the first country in the world to ban the use of ChatGPT, although that decision has in the meantime been rescinded (Schechner 2023). Restricting the use of AI-generated content has naturally led to the desire for simple detection tools. Many free online tools that claim to be able to detect AI-generated text are already available.

Some companies do urge caution when using their tools for detecting AI-generated text for taking punitive measures based solely on the results they provide. They acknowledge the limitations of their tools, e.g. OpenAI explains that there are several ways to deceive the tool (OpenAI 2023a, b, 8 May). Turnitin made a guide for teachers on how they should approach the students whose work was flagged as AI-generated

(Turnitin 2023a, b, 16 March). Nevertheless, four different companies (GoWinston, 2023; Content at Scale 2023; Compilatio 2023; GPTZero 2023) claim to be the best on the market.

The aim of this paper is to examine the general functionality of tools for the detection of the use of ChatGPT in text production, assess the accuracy of the output provided by these tools, and their efficacy in the face of the use of obfuscation techniques such as online paraphrasing tools, as well as the influence of machine translation tools to human-written text.

Specifically, the paper aims to answer the following research questions:

RQ1: Can detection tools for AI-generated text reliably detect human-written text?
RQ2: Can detection tools for AI-generated text reliably detect ChatGPT-generated text?
RQ3: Does machine translation affect the detection of human-written text?
RQ4: Does manual editing or machine paraphrasing affect the detection of ChatGPT-generated text?
RQ5: How consistent are the results obtained by different detection tools for AI-generated text?

The next section briefly describes the concept and history of LLMs. It is followed by a review of scientific and non-scientific related work and a detailed description of the research methodology. After that, the results are presented in terms of accuracy, error analysis, and usability issues. The paper ends with discussion points and conclusions made.still gained 1.0 points as in the previous methods. The formula for accuracy calculation

Large language models

We understand LLMs as systems trained to predict the likelihood of a specific character, word, or string (called a *token*) in a particular context (Bender et al. 2021). Such statistical language models have been used since the 1980s (Rosenfeld 2000), amongst other things for machine translation and automatic speech recognition. Efficient methods for the estimation of word representations in multidimensional vector spaces (Mikolov et al. 2013), together with the attention mechanism and transformer architecture (Vaswani et al. 2017) made generating human-like text not only possible, but also computationally feasible.

ChatGPT is a Natural Language Processing system that is owned and developed by OpenAI, a research and development company established in 2015. Based on the transformer architecture, OpenAI released the first version of GPT in June 2018. Within less than a year, this version was replaced by a much improved GPT-2, and then in 2020 by GPT-3 (Marr 2023). This version could generate coherent text within a given context. This was in many ways a game-changer, as it is capable of creating responses that are hard to distinguish from human-written text (Borji 2023; Brown et al. 2020). As 7% of the training data is on languages other than English, GPT-3 can also perform multilingually (Brown et al. 2020). In November 2022, ChatGPT was launched. It demonstrated significant improvements in its capabilities, a user-friendly interface, and it was widely reported in the general press. Within two months of its launch, it had over 100 million subscribers and was labelled "the fastest growing consumer app ever" (Milmo 2023).

AI in education brings both challenges and opportunities. Authorised and properly acknowledged usage of AI tools, including LLMs, is not per se a form of misconduct (Foltýnek et al. 2023). However, using AI tools in an educational context for unauthorised content generation (Foltýnek et al. 2023) **is** a form of academic misconduct (Tauginienė et al. 2018). Although LLMs have become known to the wider public after the release of ChatGPT, there is no reason to assume that they have not been used to create unauthorised and undeclared content even before that date. The accessibility, quantity, and recent development of AI tools have led many educators to demand technical solutions to help them distinguish between human-written and AI-generated texts.

For more than two decades, educators have been using software tools in an attempt to detect academic misconduct. This includes using search engines and text-matching software in order to detect instances of potential plagiarism. Although such automated detection can identify some plagiarism, previous research by Foltýnek et al. (2020) has shown that text-matching software not only do not find all plagiarism, but furthermore will also mark non-plagiarised content as plagiarism, thus providing false positive results. This is a worst-case scenario in academic settings, as an honest student can be accused of misconduct. In order to avoid such a scenario, now, when the market has responded with the introduction of dozens of tools for AI-generated text, it is important to discuss whether these tools clearly distinguish between human-written and machine-generated content.

Related work

The development of LLMs has led to an acceleration of different types of efforts in the field of automatic detection of AI-generated text. Firstly, several researchers has studied human abilities to detect machine-generated texts (e.g. Guo et al. 2023; Ippolito et al. 2020; Ma et al. 2023). Secondly, some attempts have been made to build benchmark text corpora to detect AI-generated texts effectively; for example, Liyanage et al. (2022) have offered synthetic and partial text substitution datasets for the academic domain. Thirdly, many research works are focused on developing new or fine-tuning parameters of the already pre-trained models of machine-generated text (e.g. Chakraborty et al. 2023; Devlin et al. 2019).

These efforts provide a valuable contribution to improving the performance and capabilities of detection tools for AI-generated text. In this section, the authors of the paper mainly focus on studies that compare or test the existing detection tools that educators can use to check the originality of students' assignments. The related works examined in the paper are summarised in Tables 1, 2, and 3. They are categorised as published scientific publications, preprints and other publications. It is worth mentioning that although there are many comparisons on the Internet made by individuals and organisations, Table 3 includes only those with the higher coverage of tools and/or at least partly described methodology of experiments.

Some researchers have used known text-matching software to check if they are able to find instances of plagiarism in the AI-generated text. Aydin and Karaarslan (2022) tested the iThenticate system and have revealed that the tool has found matches with

Table 1 Related work: published scientific publications

Source	Detection tools used	Dataset	Evaluation metrics
Aydin & Karaarslan 2022	1 iThenticate	An article with three sections: the text written by the paper's authors, the ChatGPT -paraphrased abstract text of articles, the content generated by ChatGPT answering specific questions	N/A
Anderson et al. 2023	1 GPT-2 Output Detector	Two ChatGPT-generated essays and the same essays paraphrased by AI	N/A
Elkhatat et al. 2023	5 OpenAI text Classifier, Writer, Copyleaks, GPTZero, CrossPlag	15 ChatGPT 3.5 generated, 15 ChatGPT 4 generated and 5 human-written passages	Specificity, Sensitivity, Positive Predictive Value, Negative Predictive Value
Gao et al. 2022	2 (Plagiarismdetector.net, GPT-2 Output Detector)	50 ChatGPT-generated scientific abstracts	AUROC

Table 2 Related work: preprints

Source	Detection tools used	Dataset	Evaluation metrics
Khalil & Er 2023	3 iThenticate, Turnitin, ChatGPT	50 essays generated by ChatGPT on various topics (such as physics laws, data mining, global warming, driving schools, machine learning, etc.)	True positive, False negative
Wang et al. 2023	6 GPT2-Detector, RoBERTa-QA, DetectGPT, GPTZero Writer, OpenAI Text Classifier	• Q&A-GPT: 115 K pairs of human-generated answers (taken from Stack Overflow) and ChatGPT generated answers (for the same topic) for 115 K questions • Code2Doc-GPT: 126 K samples from CodeSearchNet and GPT code description for 6 programming languages • 226.5 K pairs of code samples human and ChatGPT generated (APPS-GPT, CONCODE-GPT, Doc2Code-GPT) • Wiki-GPT dataset: 25 K samples of human-generated and GPT polished texts	AUC scores, False positive rate, False negative rate
Pegoraro et al. 2023	24 approaches and tools, among them online tools ZeroGPT, OpenAI Text Classifier, GPTZero, Hugging Face, Writefull, Copyleaks, Content at Scale, Originality.ai, Writer, Draft and Goal	58,546 responses generated by humans and 72,966 responses generated by the ChatGPT model, resulting in 131,512 unique samples that address 24,322 distinct questions from various fields, including medicine, opendomain, and finance	True positive rate, True negative rate

other information sources both for ChatGPT-paraphrased text and -generated text. They also found that ChatGPT does not produce original texts after paraphrasing, as the match rates for paraphrased texts were very high in comparison to human-written and

Table 3 Related work: other publications

Source	Detection tools used	Dataset	Evaluation metrics
Gewirtz 2023	3 GPT-2 Output Detector, Writer, Content at Scale	• 3 human-generated texts • 3 ChatGPT-generated texts	N/A
van Oijen 2023	7 Content at Scale, Copyleaks, Corrector App, Crossplag, GPTZero, OpenAI, Writer	• 10 generated passages based on prompts (factual info, rewrites of existing test, fictional scenarios, advice, explanations at different levels, impersonation of a specified character, Dutch translation) • 5 human-generated text from different sources (Wikipedia, SURF, Alice in Wonderland, Reddit post)	Accuracy
Compilatio 2023	11 Compilatio, Draft and Goal, GLTR, GPTZero, Content at Scale, DetectGPT, Crossplag, Kazan SEO, AI Text Classifier, Copyleaks, Writer AI Content Detector	• 50 human-written texts • 75 texts generated by ChatGPT and YouChat	Reliability (the number of correctly classified/the total number of text passages)
Demers 2023	16 Originality AI, Writer, Copyleaks, Open AI Text Classifier, Crossplag, GPTZero, Sapling, Content At Scale, Zero GPT, GLTR, Hugging Face, Corrector, Writeful, Hive Moderation, Paraphrasing tool AI Content Detector, AI Writing Check	• Human writing sample • ChatGPT 4 writing sample • ChatGPT 4 writing sample with the additional prompt "beat detection"	N/A

ChatGPT-generated text passages. In the experiment of Gao et al. (2022), Plagiarismdetector.net recognized nearly all of the fifty scientific abstracts generated by ChatGPT as completely original.

Khalil and Er (Khalil and Er 2023) fed 50 ChatGPT-generated essays into two text-matching software systems (25 essays to iThenticate and 25 essays to the Turnitin system), although they are just different interfaces to the same engine. They found that 40 (80%) of them were considered to have a high level of originality, although they defined this as a similarity score of 20% or less. Khalil and Er (Khalil and Er 2023) also attempted to test the capabilities of ChatGPT to detect if the essays were generated by ChatGPT and state an accuracy of 92%, as 46 essays were supposedly said to be cases of plagiarism. As of May 2023, ChatGPT now issues a warning to such questions such as: "As an AI language model, I cannot verify the specific source or origin of the paragraph you provided."

The authors of this paper consider the study of Khalil and Er (Khalil and Er 2023) to be problematic for two reasons. First, it is worth noting that the application of text-matching software systems to the detection of LLM-generated text makes little sense because of the stochastic nature of the word selection. Second, since an LLM will "hallucinate", that is, make up results, it cannot be asked whether it is the author of a text.

Several researchers focused on testing sets of free and/or paid detection tools for AI-generated text. Wang et al. (2023) checked the performance of detection tools on both

natural language content and programming code and determined that "detecting ChatGPT-generated code is even more difficult than detecting natural language contents." They also state that tools often exhibit bias, as some of them have a tendency to predict that content is ChatGPT generated (positive results), while others tend to predict that it is human-written (negative results).

By testing fifty ChatGPT-generated paper abstracts on the GPT-2 Output detector, Gao et al. (2022) concluded that the detector was able to make an excellent distinction between original and generated abstracts because the majority of the original abstracts were scored extremely low (corresponding to human-written content) while the detector found a high probability of AI-generated text in the majority (33 abstracts) of the ChatGPT-generated abstracts with 17 abstracts scored below 50%.

Pegoraro et al. (2023) tested not only online detection tools for AI-generated text but also many of the existing detection approaches and claimed that detection of the ChatGPT-generated text passages is still a very challenging task as the most effective online detection tool can only achieve a success rate of less than 50%. They also concluded that most of the analysed tools tend to classify any text as human-written.

Tests completed by van Oijen (2023) showed that the overall accuracy of tools in detecting AI-generated text reached only 27.9%, and the best tool achieved a maximum of 50% accuracy, while the tools reached an accuracy of almost 83% in detecting human-written content. The author concluded that detection tools for AI-generated text are "no better than random classifiers" (van Oijen 2023). Moreover, the tests provided some interesting findings; for example, the tools found it challenging to detect a piece of human-written text that was rewritten by ChatGPT or a text passage that was written in a specific style. Additionally, there was not a single attribution of a human-written text to AI-generated text, that is, an absence of false positives.

Although Demers (2023) only provided results of testing without any further analysis, their examination allows making conclusions that a text passage written by a human was recognised as human-written by all tools, while ChatGPT-generated text had a mixed evaluation with the tendency to be predicted as human-written (10 tools out of 16) that increased even further for the ChatGPT writing sample with the additional prompt "beat detection" (12 tools out of 16).

Elkhatat et al.(2023) revealed that detection tools were generally more successful in identifying GPT-3.5-generated text than GPT-4-generated text and demonstrated inconsistencies (false positives and uncertain classifications) in detecting human-written text. They also questioned the reliability of detection tools, especially in the context of investigating academic integrity breaches in academic settings.

In the tests conducted by Compilatio, the detection tools for AI-generated text detected human-written text with reliability in the range of 78–98% and AI-generated text – 56–88%. Gewirtz' (2023) results on testing three human-written and three ChatGPT-generated texts demonstrated that two of the selected detection tools for AI-generated text could reach only 50% accuracy and one an accuracy of 66%.

The effect of paraphrasing on the performance of detection tools for AI-generated text has also been studied. For example, Anderson et al. (2023) concluded that paraphrasing has significantly lowered the detection capabilities of the GPT-2 Output Detector by increasing the score for human-written content from 0.02% to 99.52% for the first essay

and from 61.96% to 99.98% for the second essay. Krishna et al. (2023) applied paraphrasing to the AI-generated texts and revealed that it significantly lowered the detection accuracy of five detection tools for AI-generated text used in the experiments.

The results of the above-mentioned studies suggest that detecting AI-generated text passages is still challenging for existent detection tools for AI-generated text, whereas human-written texts are usually identified quite accurately (accuracy above 80%). However, the ability of tools to identify AI-generated text is under question as their accuracy in many studies was only around 50% or slightly above. Depending on the tool, a bias may be observed identifying a piece of text as either ChatGPT-generated or human-written. In addition, tools have difficulty identifying the source of the text if ChatGPT transforms human-written text or generates text in a particular style (e.g. a child's explanation). Furthermore, the performance of detection tools significantly decreases when texts are deliberately modified by paraphrasing or re-writing. Detection of the AI-generated text remains challenging for existing detection tools, but detecting ChatGPT-generated code is even more difficult.

Existing research has several shortcomings:

- quite often experiments are carried out with a limited number of detection tools for AI-generated text on a limited set of data;
- sometimes human-written texts are taken from publicly available websites or recognised print sources, and thus could potentially have been previously used to train LLMs and/or provide no guarantee that they were actually written by humans;
- the methodological aspects of the research are not always described in detail and are thus not available for replication;
- testing whether the AI-generated and further translated text can influence the accuracy of the detection tools is not discussed at all;
- a limited number of measurable metrics is used to evaluate the performance of detection tools, ignoring the qualitative analysis of results, for example, types of classification errors that can have significant consequences in an academic setting.

Methodology

Test cases

The focus of this research is determining the accuracy of tools which state that they are able to detect AI-generated text. In order to do so, a number of situational parameters were set up for creating the test cases for the following categories of English-language documents:

- human-written;
- human-written in a non-English language with a subsequent AI/machine translation to English;
- AI-generated text;
- AI-generated text with subsequent human manual edits;
- AI-generated text with subsequent AI/machine paraphrase.

For the first category (called 01-Hum), the specification was made that 10.000 characters (including spaces) were to be written at about the level of an undergraduate in the field of the researcher writing the paper. These fields include academic integrity, civil engineering, computer science, economics, history, linguistics, and literature. None of the text may have been exposed to the Internet at any time or even sent as an attachment to an email. This is crucial because any material that is on the Internet is potentially included in the training data for an LLM.

For the second category (called 02-MT), around 10.000 characters (including spaces) were written in Bosnian, Czech, German, Latvian, Slovak, Spanish, and Swedish. None of this texts may have been exposed to the Internet before, as for 01-Hum. Depending on the language, either the AI translation tool DeepL (3 cases) or Google Translate (6 cases) was used to produce the test documents in English.

It was decided to use ChatGPT as the only AI-text generator for this investigation, as it was the one with the largest media attention at the beginning of the research. Each researcher generated two documents with the tool using different prompts, (03-AI and 04-AI) with a minimum of 2000 characters each and recorded the prompts. The language model from February 13, 2023 was used for all test cases.

Two additional texts of at least 2000 characters were generated using fresh prompts for ChatGPT, then the output was manipulated. It was decided to use this type of test case, as students will have a tendency to obfuscate results with the expressed purpose of hiding their use of an AI-content generator. One set (05-ManEd) was edited manually with a human exchanging some words with synonyms or reordering sentence parts and the other (06-Para) was rewritten automatically with the AI-based tool Quillbot (Quillbot 2023), using the default values of the tool for modes (Standard) and synonym level. Documentation of the obfuscation, highlighting the differences between the texts, can be found in the Appendix.

With nine researchers preparing texts (the eight authors and one collaborator), 54 test cases were thus available for which the ground truth is known.

AI-generated text detection tool selection

A list of detection tools for AI-generated text was prepared using social media and Google search. Overall, 18 tools were considered, out of which 6 were excluded: 2 were not available, 2 were not online applications but Chrome extensions and thus out of the scope of this research, 1 required payment, and 1 did not produce any quantifiable result.

The company Turnitin approached the research group and offered a login, noting that they could only offer access from early April 2023. It was decided to test the system, although it is not free, because it is so widely used and already widely discussed in academia. Another company, PlagiarismCheck, was also advertising that it had a detection tool for AI-generated text in addition to its text-matching detection system. It was decided to ask them if they wanted to be part of the test as well, as the researchers did not want to have only one paid system. They agreed and provided a login in early May. We caution that their results may be different from the free tools used, as the companies knew that the submitted documents were part of a test suite and they were able to use the entire test document.

The following 14 detection tools were tested:

- Check For AI (https://checkforai.com)
- Compilatio (https://ai-detector.compilatio.net/)
- Content at Scale (https://contentatscale.ai/ai-content-detector/)
- Crossplag (https://crossplag.com/ai-content-detector/)
- DetectGPT (https://detectgpt.ericmitchell.ai/)
- Go Winston (https://gowinston.ai)
- GPT Zero (https://gptzero.me/)
- GPT-2 Output Detector Demo (https://openai-openai-detector.hf.space/)
- OpenAI Text Classifier (https://platform.openai.com/ai-text-classifier)
- PlagiarismCheck (https://plagiarismcheck.org/)
- Turnitin (https://demo-ai-writing-10.turnitin.com/home/)
- Writeful GPT Detector (https://x.writefull.com/gpt-detector)
- Writer (https://writer.com/ai-content-detector/)
- Zero GPT (https://www.zerogpt.com/)

Table 4 gives an overview of the minimum/maximum sizes of text that could be examined by the free tools at the time of testing, if known.

PlagiarismCheck and Turnitin are combined text similarity detectors and offer an additional functionality of determining the probability the text was written by an AI, so there was no limit on the amount of text tested. Signup was necessary for Check for AI, Crossplag, Go Winston, GPT Zero, and OpenAI Text Classifier (a Google account worked).

Data collection

The tests were run by the individual authors between March 7 and March 28, 2023. Since Turnitin was not available until April, those tests were completed between April 14 and April 20, 2023. The testing of PlagiarismCheck was performed between May 2

Table 4 Minimum and maximum sizes for free tools

Tool name	Minimum Size	Maximum Size
Check for AI	350 characters	2500 characters
Compilatio	200 characters	2000 characters
Content at Scale	25 words	25000 characters
Crossplag	Not stated	1000 words
DetectGPT	40 words	256 words
Go Winston	500 characters	2000 words
GPT Zero	250 characters	5000 characters
GPT-2 Output Detector Demo	50 tokens	510 tokens
OpenAI Text Classifier	1000 characters	Not stated
Writeful GPT Detector	50 words	1000 words
Writer	Not stated	1500 characters
Zero GPT	Not stated	Not stated

Table 5 Classification accuracy scales for human-written and AI-generated texts

Human-written (NEGATIVE) text (docs 01-Hum & 02-MT), and the tool says that it is written by a:		
[100—80%) human	True negative	TN
[80—60%) human	Partially true negative	PTN
[60—40%) human	Unclear	UNC
[40—20%) human	Partially false positive	PFP
[20—0%] human	False positive	FP
AI-generated (POSITIVE) text (docs 03-AI, 04-AI, 05-ManEd & 06-Para), and the tool says it is written by a:		
[100—80%) human	False negative	FN
[80—60%) human	Partially false negative	PFN
[60—40%) human	Unclear	UNC
[40—20%) human	Partially true positive	PTP
[20—0%] human	True positive	TP

[or] means inclusive (or) means exclusive

Table 6 Mapping of textual results to classification labels

Tool	Result	01-Hum, 02-MT	03-AI, 04-AI, 05-ManEd, 06-Para
Check for AI	"very low risk"	TN	FN
	"low risk"	PTN	PFN
	"medium risk"	UNC	UNC
	"high risk"	PFP	PTP
	"very high risk"	FP	TP
GPT Zero	"likely to be written entirely by human"	TN	FN
	"may include parts written by AI"	PFP	PTP
	"likely to be written entirely by AI"	FP	TP
OpenAI Text Classifier	"The classifier considers the text to be …"		
	"… likely AI-generated."	FP	TP
	"… possibly AI-generated."	PFP	PTP
	"Unclear if it is AI-generated"	UNC	UNC
	"… unlikely AI-generated."	PTN	PFN
	"… very unlikely AI-generated."	TN	FN
DetectGPT	"very unlikely to be from GPT-2"	TN	FN
	"unlikely to be from GPT-2"	PTN	PFN
	"likely to be from GPT-2"	PFP	PTP
	"very likely from GPT-2"	FP	TP

and May 8, 2023. All the 54 test cases had been presented to each of the tools for a total of 756 tests.

Evaluation methodology

For the evaluation, the authors were split into groups of two or three and tasked with evaluating the results of the tests for the cases from either 01-Hum & 04-AI, 02-MT & 05-ManEd, or 03-AI & 06-Para. Since the tools do not provide an exact binary classification, one five-step classification was used for the original texts (01-Hum & 02-MT)

and another one was used for the AI-generated texts (03-AI, 04-AI, 05-ManEd & 06-Para). They were based on the probabilities that were reported for texts being human-written or AI-generated as specified in Table 5.

For four of the detection tools, the results were only given in the textual form ("very low risk", "likely AI-generated", "very unlikely to be from GPT-2", etc.) and these were mapped to the classification labels as given in Table 6.

After all of the classifications were undertaken and disagreements ironed out, the measures of accuracy, the false positive rate, and the false negative rate were calculated.

Results

Having evaluated the classification outcomes of the tools as (partially) true/false positives/negatives, the researchers evaluated this classification on two criteria: accuracy and error type. In general, classification systems are evaluated using accuracy, precision, and recall. The research authors also conducted an error analysis since the educational context means different types of error have different significance.

Accuracy

When no partial results are allowed, i.e. only TN, TP, FN, and FP are allowed, accuracy is defined as a ratio of correctly classified cases to all cases

$$\mathrm{ACC} = (\mathrm{TN} + \mathrm{TP})/(\mathrm{TN} + \mathrm{TP} + \mathrm{FN} + \mathrm{FP});$$

As our classificaion contains also partially correct and partially incorrect results (i.e., five classes instead of two), the basic commonly used formula has to be adjusted to properly count these cases. There is no standard way of how this adjustment should be done. Therefore, we will use three different methods which we believe reflect different approaches that educators may have when interpreting tools' outputs. The first (binary)

Table 7 Accuracy of the detection tools (binary approach)

Tool	01-Hum	02-MT	03-AI	04-AI	05-ManEd	06-Para	Total	Accuracy	Rank
Check For AI	9	0	9	8	4	2	32	59%	6
Compilatio	8	9	8	8	5	2	40	74%	2
Content at Scale	9	9	0	0	0	0	18	33%	14
Crossplag	9	6	9	7	4	2	37	69%	4
DetectGPT	9	5	2	8	0	1	25	46%	11
Go Winston	7	7	9	8	4	1	36	67%	5
GPT Zero	6	3	7	7	3	3	29	54%	8
GPT-2 Output Detector Demo	9	7	9	8	5	1	39	72%	3
OpenAI Text Classifier	9	8	2	7	2	1	29	54%	8
PlagiarismCheck	7	5	3	3	1	2	21	39%	13
Turnitin	9	9	8	9	4	2	41	76%	1
Writeful GPT Detector	9	7	2	3	2	0	23	43%	12
Writer	9	7	4	4	2	1	27	50%	10
Zero GPT	9	5	7	8	2	1	32	59%	6
Average	**94%**	**69%**	**63%**	**70%**	**30%**	**15%**			

approach is to consider partially correct classification as incorrect and calculate the accuracy as

$$\text{ACC\_bin} = (\text{TN} + \text{TP})/(\text{TN} + \text{PTN} + \text{TP} + \text{PTP} + \text{FN} + \text{PFN} + \text{FP} + \text{PFP} + \text{UNC})$$

For the systems providing percentages of confidence, this method basically sets the threshold of 80% (see Table 5). Table 7 shows the number of correctly classified documents, i.e. the sum of true positives and true negatives. The maximum for each cell is 9 (because there were 9 documents in each class), the overall maximum is 9 * 6 = 54. The accuracy is calculated as a ratio of the total and the overall maximum. Note that even the highest accuracy values are below 80%. The last row shows the average accuracy for each document class, across all the tools.

This method provides a good overview of the number of cases in which the classifiers are "sure" about the outcome. However, for real-life educational scenarios, partially correct classifications are also valuable. Especially in case 05-ManEd, which involved human editing, the partially positive classification results make sense. Therefore, the researchers explored more ways of assessment. These methods differ in the score awarded to various incorrect outcomes.

In our second approach, we include partially correct evaluations and count them as correct ones. The formula for accuracy computation is.

$$\text{ACC\_bin\_incl} = (\text{TN}+\text{PTN}+\text{TP}+\text{PTP})/(\text{TN}+\text{PTN}+\text{TP}+\text{PTP}+\text{FN}+\text{PFN}+\text{FP}+\text{PFP}+\text{UNC})$$

In case of systems providing percentages, this method basically sets the threshold of 60% (see Table 5). The results of this classification approach may be found in Table 8. Obivously, all systems achieved higher accuracy, and the systems that provided more partially correct results (GPT Zero, Check for AI) influenced the order.

In our third approach, which we call semi-binary evaluation, the researchers distinguish partially correct classifications (PTN or PTP) both from the correct and incorrect ones. The partially correct classifications were awarded 0.5 points, while entirely correct

Table 8 Accuracy of the detection tools (binary inclusive approach)

Tool	01-Hum	02-MT	03-AI	04-AI	05-ManEd	06-Para	Total	Accuracy	Rank
Check For AI	9	7	9	8	4	3	40	74%	4
Compilatio	9	9	9	8	6	2	43	80%	2
Content at Scale	9	9	0	0	0	0	18	33%	14
Crossplag	9	6	9	7	5	2	38	70%	9
DetectGPT	9	8	9	8	4	2	40	74%	4
Go Winston	8	8	9	8	5	2	40	74%	4
GPT Zero	6	3	8	9	8	8	42	78%	3
GPT-2 Output Detector Demo	9	7	9	8	5	2	40	74%	4
OpenAI Text Classifier	9	9	5	8	5	2	38	70%	9
PlagiarismCheck	9	8	5	6	3	3	34	63%	12
TurnItIn	9	9	9	9	5	3	44	81%	1
Writeful GPT Detector	9	8	8	6	3	1	35	65%	11
Writer	9	7	5	6	4	2	33	61%	13
Zero GPT	9	8	7	8	4	4	40	74%	4

Table 9 Accuracy of the detection tools (semi-binary approach)

Tool	01-Hum	02-MT	03-AI	04-AI	05-ManEd	06-Para	Total	Accuracy	Rank
Check For AI	9	3.5	9	8	4	2.5	36	67%	6
Compilatio	8.5	9	8.5	8	5.5	2	41.5	77%	2
Content at Scale	9	9	0	0	0	0	18	33%	14
Crossplag	9	6	9	7	4.5	2	37.5	69%	5
DetectGPT	9	6.5	5.5	8	2	1.5	32.5	60%	10
Go Winston	7.5	7.5	9	8	4.5	1.5	38	70%	4
GPT Zero	6	3	7.5	8	5.5	5.5	35.5	66%	8
GPT-2 Output Detector Demo	9	7	9	8	5	1.5	39.5	73%	3
OpenAI Text Classifier	9	8.5	3.5	7.5	3.5	1.5	33.5	62%	9
PlagiarismCheck	8	6.5	4	4.5	2	2.5	27.5	51%	13
Turnitin	9	9	8.5	9	4.5	2.5	42.5	79%	1
Writeful GPT Detector	9	7.5	5	4.5	2.5	0.5	29	54%	12
Writer	9	7	4.5	5	3	1.5	30	56%	11
Zero GPT	9	6.5	7	8	3	2.5	36	67%	6
Average	**95%**	**77%**	**71%**	**74%**	**39%**	**22%**			

Table 10 Scores for logarithmic evaluation

Positive case	Negative case	Score
FN	FP	1
PFN	PFP	2
UNC	UNC	4
PTP	PTN	8
TP	TN	16

classification (TN or TP) still gained 1.0 points as in the previous methods. The formula for accuracy calculation is

$$\begin{aligned} \text{ACC\_semibin} = &(\text{TN} + \text{TP} + 0.5 * \text{PTN} + 0.5 * \text{PTP}) / \\ &(\text{TN} + \text{PTN} + \text{TP} + \text{PTPFN} + \text{PFN} + \text{FP} + \text{PFP} + \text{UNC}) \end{aligned}$$

Table 9 shows the assessment results of the classifiers using semi-binary classification. The values correspond to the number of correctly classified documents with partially correct results awarded half a point (TP + TN + 0.5 * PTN + 0.5 * PTP). The maximum value is again 9 for each cell and 54 for the total.

A semi-binary approach to accuracy calculation captures the notion of partially correct classification but still does not distinguish between various forms of incorrect classification. We address this issue by employing a third,—logarithmic approach to accuracy calculation that awards 1 point to completely incorrect classification and doubles the score for each level of the classification that was closer to the correct result. The scores for the particular classifier outputs are shown in Table 10 and the overall scores of the classifiers are shown in Table 11. Note that the maximum value for each cell is now 9 * 16 = 864. The accuracy, again, is calculated as a ratio

Table 11 Logarithmic approach to accuracy evaluation

Tool	01-Hum	02-MT	03-AI	04-AI	05-ManEd	06-Para	Total	Accuracy	Rank
Check For AI	144	62	144	129	74	54	607	70%	7
Compilatio	136	144	136	132	91	40	679	79%	2
Content at Scale	144	144	23	24	17	18	370	43%	14
Crossplag	144	99	144	115	76	40	618	72%	6
DetectGPT	144	108	88	129	38	36	543	63%	10
Go Winston	124	124	144	130	79	45	646	75%	4
GPT Zero	102	60	121	128	89	89	589	68%	8
GPT-2 Output Detector Demo	144	114	144	129	84	35	650	75%	3
OpenAI Text Classifier	144	136	67	124	67	48	586	68%	9
PlagiarismCheck	128	108	76	82	50	53	497	58%	12
Turnitin	144	144	136	144	81	53	702	81%	1
Writeful GPT Detector	144	122	81	76	50	20	493	57%	13
Writer	144	117	83	84	53	35	516	60%	11
Zero GPT	144	108	120	132	65	54	623	72%	5
Average	**96%**	**79%**	**75%**	**77%**	**45%**	**31%**			

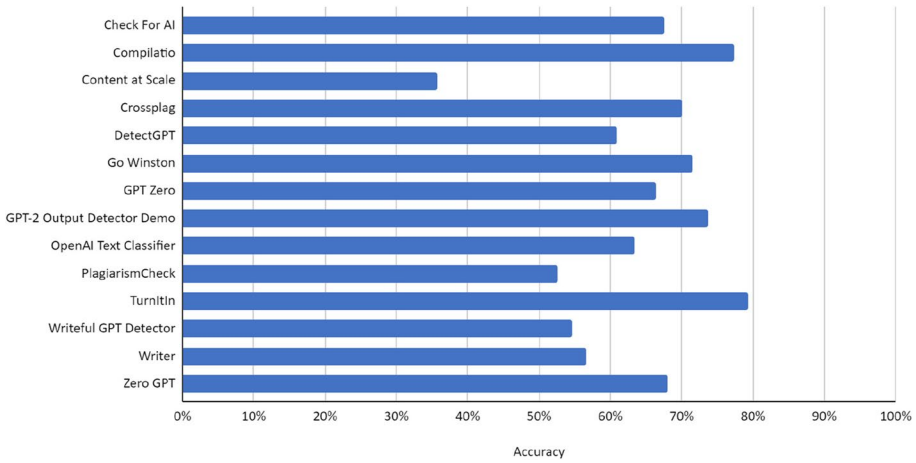


Fig. 1 Overall accuracy for each tool calculated as an average of all approaches discussed

of the total score and the maximum possible score. This approach provides the most detailed distinction among all varieties of (in)correctness.

As can be seen from Tables 7, 8, 9, and 11, the approach to accuracy evaluation has almost no influence on the ranking of the classifiers. Figure 1 presents the overall accuracy for each tool as the mean of all accuracy approaches used.

Turnitin received the highest score using all approaches to accuracy classification, followed by Compilatio and GPT-2 Output Detector (again in all approaches). This is particularly interesting because as the name suggests, GPT-2 Output Detector was not trained to detect GPT-3.5 output. Crossplag and Go Winston were the only other tools to achieve at least 70% accuracy.

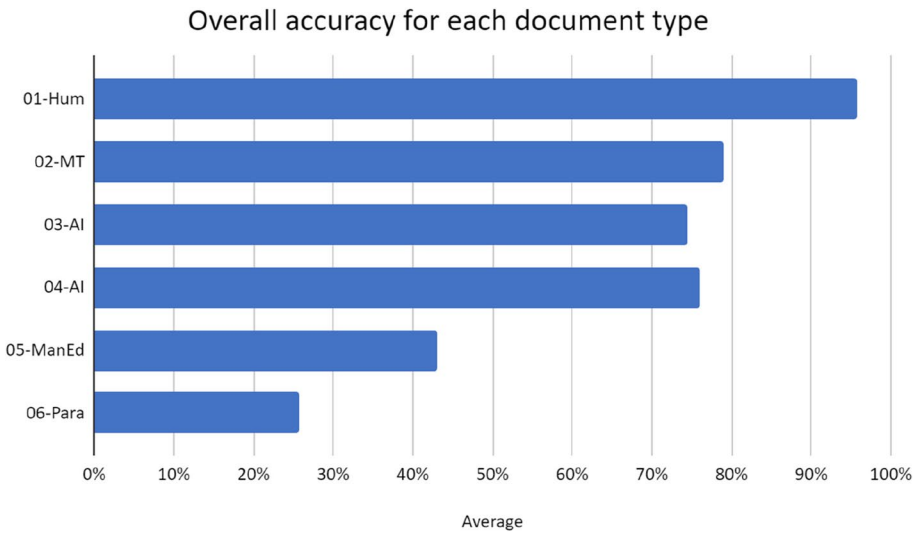


Fig. 2 Overall accuracy for each document type (calculated as an average of all approaches discussed)

Variations in accuracy

As Fig. 2 above shows, the overall average accuracy figure is misleading, as it obscures major variations in accuracy between document types. Further analysis reveals the influence of machine translation, human editing, and machine paraphrasing on overall accuracy:

Influence of machine translation The overall accuracy for case 01-Hum (human-written) was 96%. However, in the case of the documents written by humans in languages other than English that were machine-translated to English (case 02-MT), the accuracy dropped by 20%. Apparently, machine translation leaves some traces of AI in the output, even if the original was purely human-written.

Influence of human manual editing Case 05-ManEd (machine-generated with subsequent human editing) generally received slightly over half the score (42%) compared to cases 03-AI and 04-AI (machine-generated with no further modifications; 74%). This reflects a typical scenario of student misconduct in cases where the use of AI is prohibited. The student obtains a text written by an AI and then quickly goes through it and makes some minor changes such as using synonyms to try to disguise unauthorised content generation. This type of writing has been called patchwriting (Howard 1995). Only ~ 50% accuracy of the classifiers shows that these cases, which are assumed to be the most common ones, are almost undetectable by current tools.

Influence of machine paraphrase Probably the most surprising results are for case 06-Para (machine-generated with subsequent machine paraphrase). The use of AI to transform AI-generated text results in text that the classifiers consider human-written. The overall accuracy for this case was 26%, which means that most AI-generated texts remain undetected when machine-paraphrased.

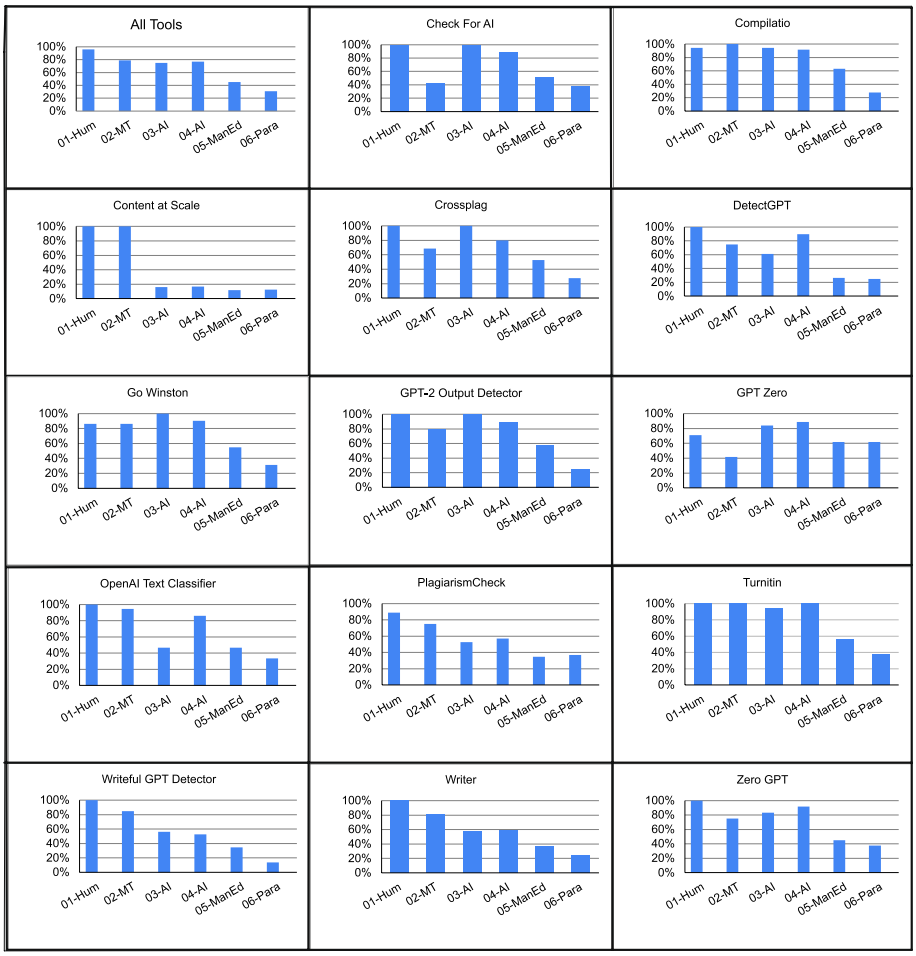


Fig. 3 Accuracy (logarithmic) for each document type by detection tool for AI-generated text

Consistency in tool results

With the notable exception of GPT Zero, all the tested tools followed the pattern of higher accuracy when identifying human-written text than when identifying texts generated or modified by AI or machine tools, as seen in Fig. 3. Therefore, their classification is (probably deliberately) biased towards humans rather than AI output. This classification bias is preferable in academic contexts for the reasons discussed below.

Precision

Another important indicator of system's performance is precision, i.e. the ratio of true positive cases to all positively classified cases. Precision indicates the probability that a positive classification provided by the system is correct. For pure binary classifiers, the precision is calculated as a ratio of true positives to all positively classified cases:

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

Table 12 Overview of classification results and precision

Tool	TP	PTP	FP	PFP	TN	PTN	FN	PFN	UNC	Total	Prec_incl	Prec_excl
Check For AI	23	1		1	9	7	1	10	2	54	96%	100%
Compilatio	23	2			17	1	9	1	1	54	100%	100%
Content at Scale					18		12	13	11	54	–-	–-
Crossplag	22	1	3		15		11	2		54	88%	88%
DetectGPT	11	12			14	3	7	6	1	54	100%	100%
Go Winston	22	2			14	2	4	3	7	54	100%	100%
GPT Zero	20	13		9	9		3			54	79%	100%
GPT-2 Output Detector Demo	23	1	2		16		10	1	1	54	92%	92%
OpenAI Text Classifier	12	8			17	1	2	4	10	54	100%	100%
PlagiarismCheck	9	8			12	5	1	10	9	54	100%	100%
TurnItIn	23	3			18		4	3	3	54	100%	100%
Writeful GPT Detector	7	11		1	16	1	13	3	2	54	95%	100%
Writer	11	6	1		16		13	3	4	54	94%	92%
Zero GPT	18	5			14	3	3		11	54	100%	100%

In case of partially true/false positives, the researches had two options how to deal with them. The exclusive approach counts them as negatively classified (so the formula does not change), whereas the inclusive approach counts them as positively classified:

$$\text{Precision\_incl} = (\text{TP} + \text{PTP})/(\text{TP} + \text{PTP} + \text{FP} + \text{PFP})$$

Table 12 shows an overview of the classification results, i.e. all (partially) true/false positives/negatives. Also, both inclusive and exclusive precision values are provided. Precision is missing for Content at Scale because this system did not provide any positive classifications. The only system for which the inclusive precision is significantly different from the exclusive one, is GPT Zero which yielded the largest number of partially false positives.

Error analysis

In this section, the researchers quantify more indicators of tools' performance, namely two types of classification errors that might have significant consequences in educational contexts: false positives leading to false accusations against a student and undetected cases (students gaining an unfair advantage over others), i.e. false negative ratio which is tightly related to recall.

False accusations: harm to individual students

If educators use one of the classifiers to detect student misconduct, there is a question of what kind of output leads to the accusation of a student from unauthorised content generation. The researchers believe that a typical educator would accuse a student if the output of the classifier is positive or partially positive. Some teachers may also suspect students of misconduct in unclear or partially negative cases, but the research authors think that educators generally do not initiate disciplinary action in these cases.

Table 13 False positive (false accusation) ratio

Tool	01-Hum	02-MT	Total	FPR
Check For AI	0	1	1	5.6%
Compilatio	0	0	0	0.0%
Content at Scale	0	0	0	0.0%
Crossplag	0	3	3	16.7%
DetectGPT	0	0	0	0.0%
Go Winston	0	0	0	0.0%
GPT Zero	3	6	9	50.0%
GPT-2 Output Detector Demo	0	2	2	11.1%
OpenAI Text Classifier	0	0	0	0.0%
PlagiarismCheck	0	0	0	0.0%
Turnitin	0	0	0	0.0%
Writeful GPT Detector	0	1	1	5.6%
Writer	0	1	1	5.6%
Zero GPT	0	0	0	0.0%
Average	**2.4%**	**11.1%**		

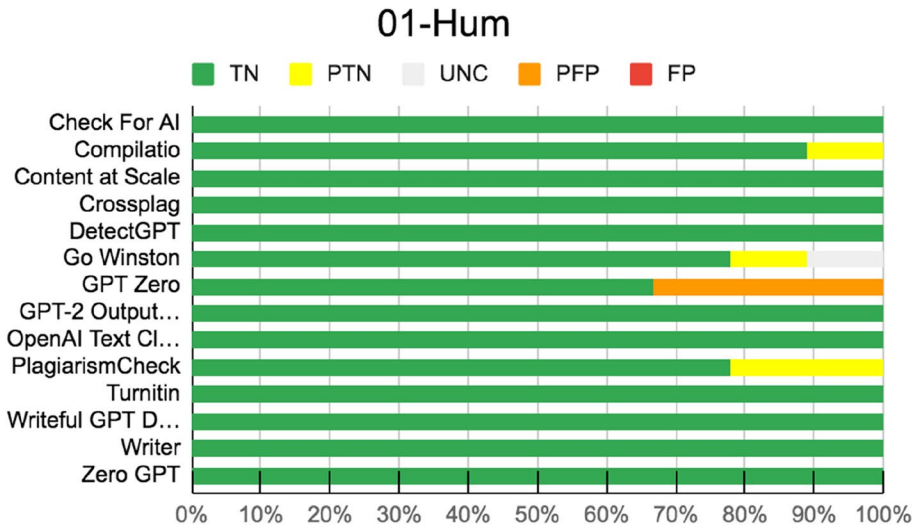


Fig. 4 False accusations for human-written documents

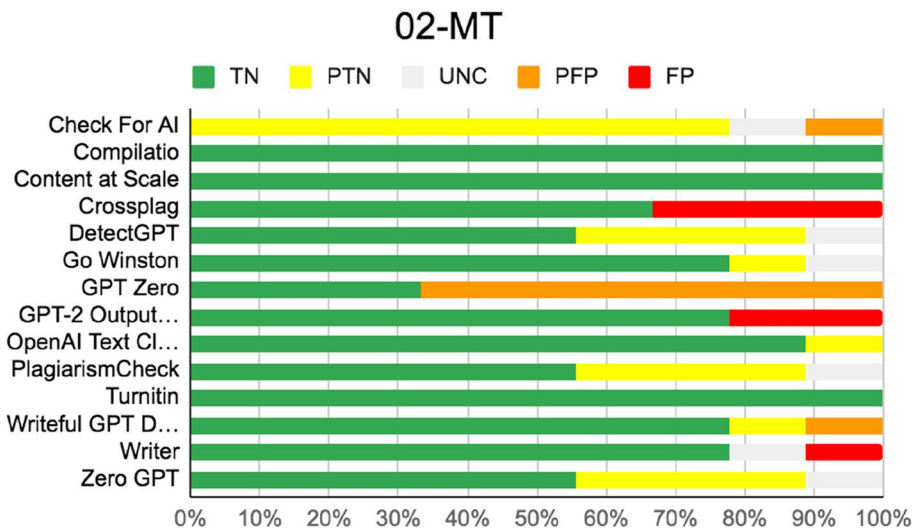


Fig. 5 False accusations for machine-translated documents

Therefore, for each tool, we also computed the likelihood of false accusation of a student as a ratio of false positives and partially false positives to all negative cases, i.e.

$$\mathrm{FPR} = (\mathrm{FP} + \mathrm{PFP})/\mathrm{N\_negative}$$

Table 13 shows the number of cases in which the classification of a particular document would lead to a false accusation. The table includes only documents 01-Hum and 02-MT, because the AI-generated documents are not relevant. The risk of false accusations is zero for half of the tools, as can be also seen from Figs. 4 and 5. Six of the fourteen tools tested generated false positives, with the risk increasing dramatically for machine-translated texts. For GPT Zero, half of the positive classifications would be false accusations, which makes this tool unsuitable for the academic environment.

Table 14 Percentage of undetected cases

Tool	03-AI	04-AI	05-ManEd	06-Para	Total	FNR	Recall
Check For AI	0	1	5	6	12	33.3%	66.7%
Compilatio	0	1	3	7	11	30.6%	69.4%
Content at Scale	9	9	9	9	36	100.0%	0.0%
Crossplag	0	2	4	7	13	36.1%	63.9%
DetectGPT	0	1	5	7	13	36.1%	63.9%
Go Winston	0	1	4	7	12	33.3%	66.7%
GPT Zero	1	0	1	1	3	8.3%	91.7%
GPT-2 Output Detector Demo	0	1	4	7	12	33.3%	66.7%
OpenAI Text Classifier	4	1	4	7	16	44.4%	55.6%
PlagiarismCheck	4	3	6	6	19	52.8%	47.2%
Turnitin	0	0	4	6	10	27.8%	72.2%
Writeful GPT Detector	1	3	6	8	18	50.0%	50.0%
Writer	4	3	5	7	19	52.8%	47.2%
Zero GPT	2	1	5	5	13	36.1%	63.9%
Average	**19.8%**	**21.4%**	**51.6%**	**71.4%**			

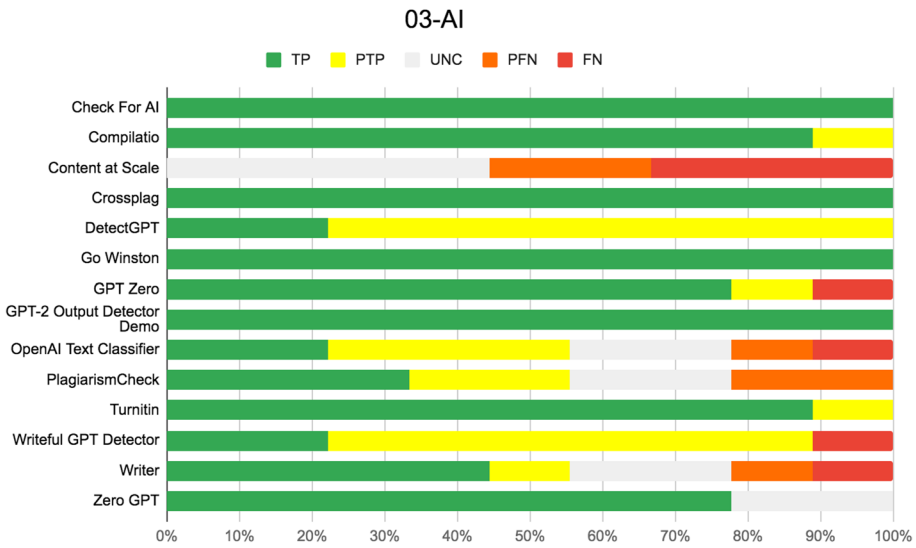


Fig. 6 False negatives for AI-generated documents 03-AI

Undetected cases: undermining academic integrity

Another form of academic harm is undetected cases, i.e. AI-generated texts that remain undetected. A student who used unauthorised content generation likely obtains an unfair advantage over those who fulfilled the task with integrity. The actual victims of this form of misconduct are the honest students that receive the same credits as the dishonest ones. The likelihood of an AI-generated document being undetected (false negative rate, FNR) is given in Table 14, which includes only positive cases (03-AI, 04-AI, 05-ManEd and 06-Para). The false negative rate is calculated as

$$\mathrm{FNR} = (\mathrm{FN} + \mathrm{PFN})/\mathrm{N\_positive}$$

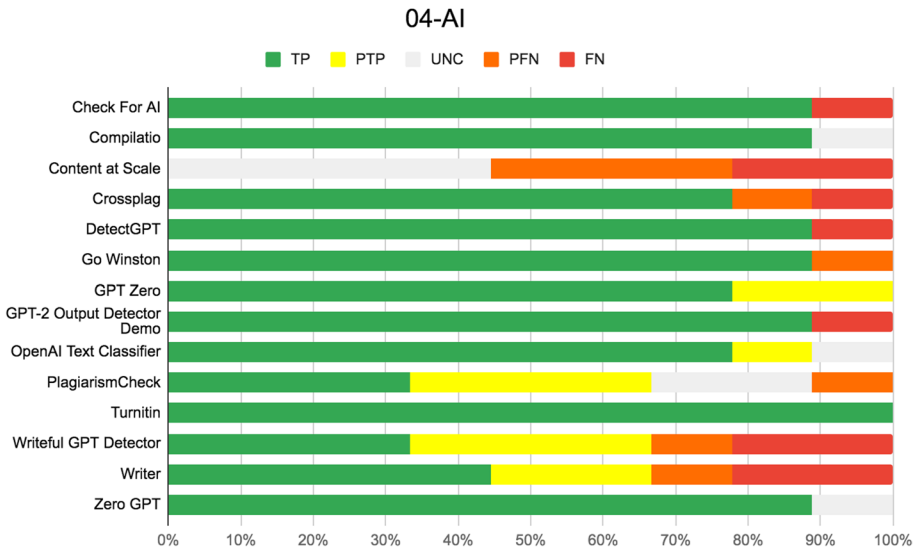


Fig. 7 False negatives for AI-generated documents 04-AI

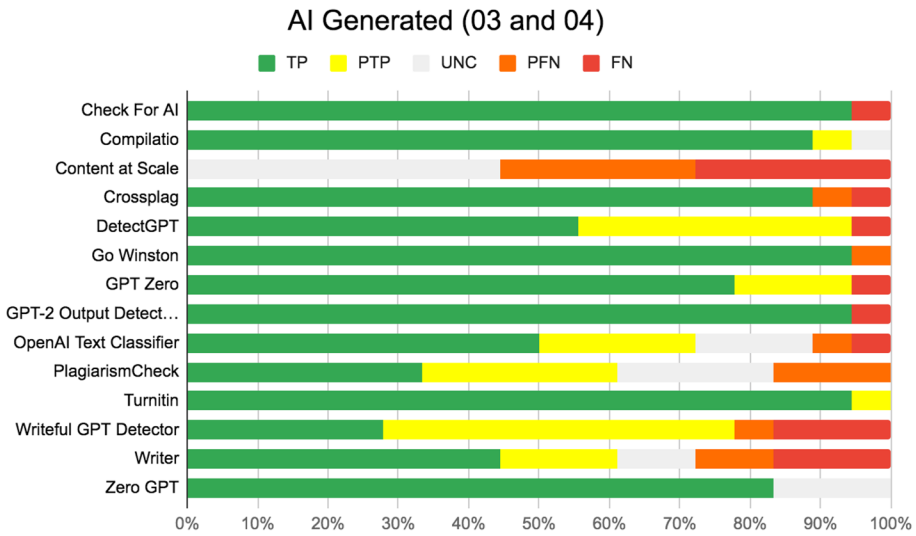


Fig. 8 False negatives for AI-generated documents 03-AI and 04-AI together

For the sake of completeness, Table 14 also contains recall (1—FNR) that indicates how many of positive cases were correclty classified by the system.

Figures 6, 7, and 8 above show that 13 out of the 14 tested tools produced false negatives or partially false negatives for documents 03-AI and 04-AI; only Turnitin correctly classified all documents in these classes. None of the tools could correctly classify all AI-generated documents that undergo manual editing or machine paraphrasing.

As the document sets 03-AI and 04-AI were prepared using the same method, the researchers expected the results would be the same. However, for some tools (OpenAI Text Classifier and DetectGPT), the results were notably different. This could indicate a mistake in testing made or interpretation of the results. Therefore, the researchers

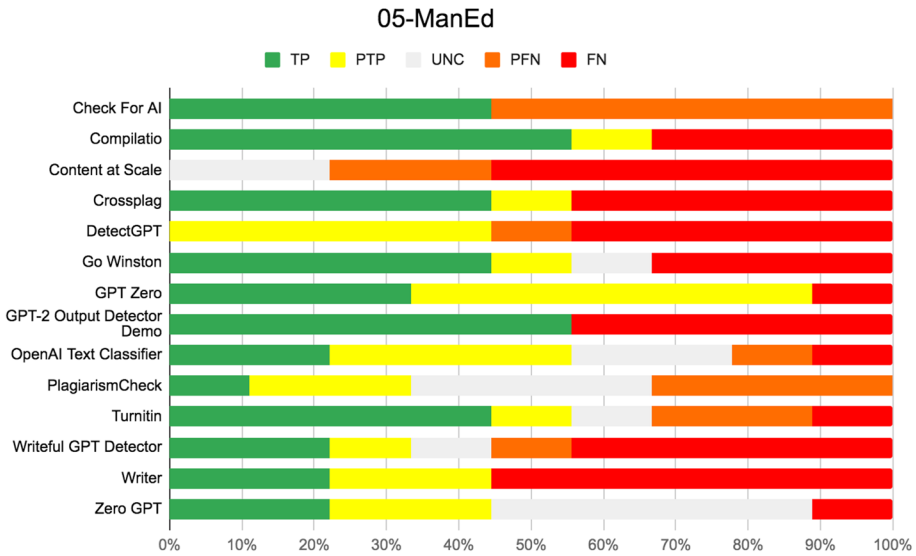


Fig. 9 False negatives for manually edited documents

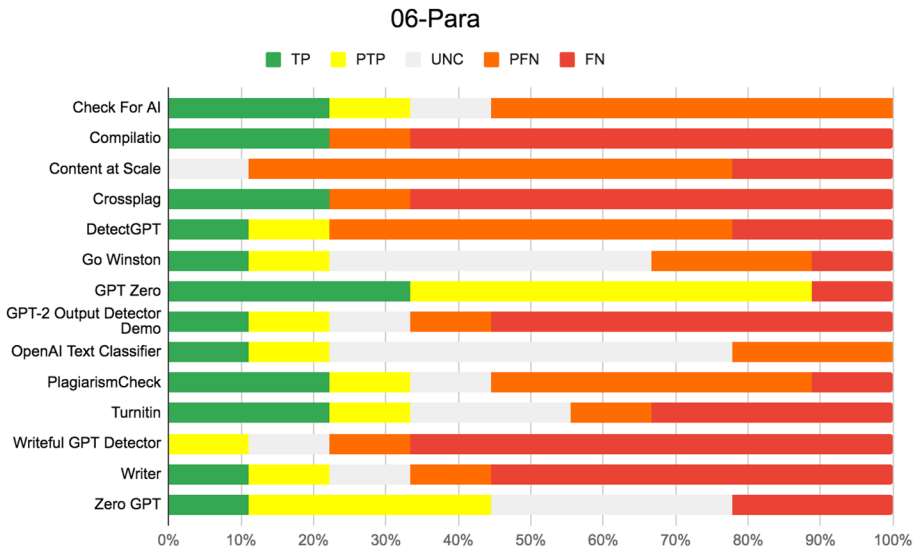


Fig. 10 False negatives for machine-paraphrased documents

double-checked all the results to avoid this kind of mistake. We also tried to upload some documents again. We did obtain different values, but we found out that this was due to inconsistency in the results of these tools and not due to our mistakes.

Content at Scale misclassified all of the positive cases; these results in combination with the 100% correct classification of human-written documents indicate that the tool is inherently biased towards human classification and thus completely useless. Overall, of the AI-generated texts approx. 20% of cases would likely be misattributed to humans, meaning the risk of unfair advantage is significantly greater than that of false accusation.

Figures 9 and 10 show an even greater risk of students gaining an unfair advantage through the use of obfuscation strategies. At an overall level, for manually edited texts

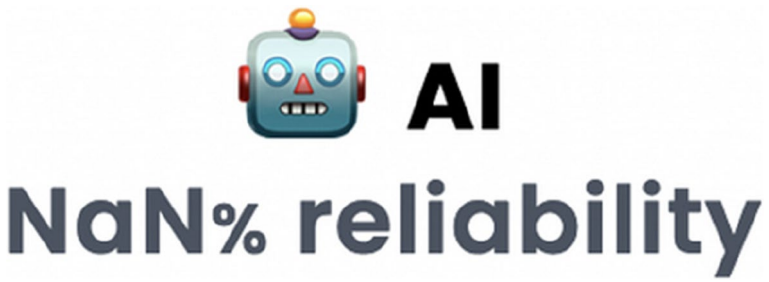


Fig. 11 Compilatio's NaN% reliability

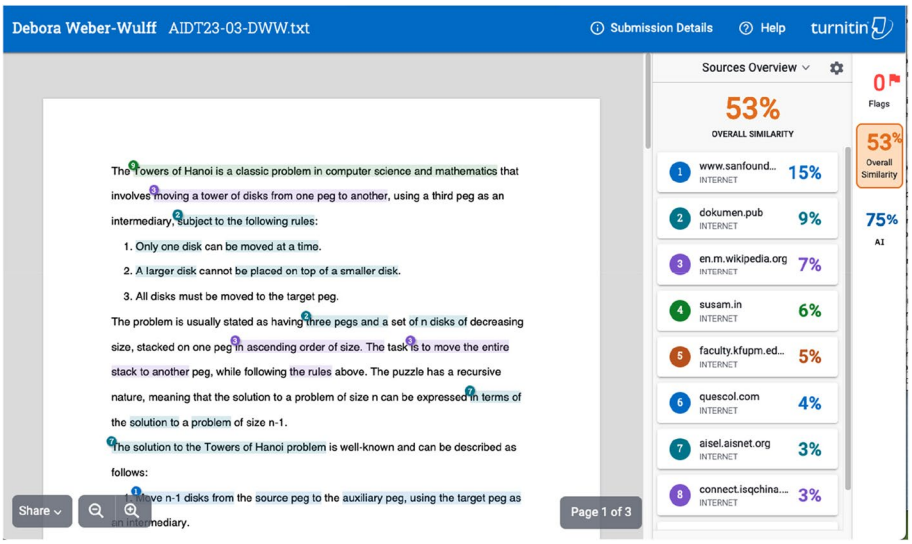


Fig. 12 Turnitin's similarity report shows up first, it is not clear that the "AI" is clickable

(case 05-ManEd) the ratio of undetected texts increases to approx. 50% and in the case of machine-paraphrased texts (case 06-Para) rises even higher.

Usability issues

There were a few usability issues that cropped up during the testing that may be attributable to the beta nature of the tools under investigation.

For example, the tool DetectGPT at some point stopped working and only replied with the statement "Server error 😕 We might just be overloaded. Try again in a few minutes?". This issue occurred after the initial testing round and persisted until the time of submission of this paper. Others would stall in an apparent infinite loop or throw an error message and the test had to be repeated at a later time.

Writeful GPT Detector would not accept computer code. The tool apparently identified code as not English, and the tool only accepted English texts.

Compilatio at one point returned "NaN% reliability" (See Fig. 11) for a ChatGPT-generated text that included program code. "NaN" is computer jargon for "not a

number" and indicates that there were calculation issues such as division by zero or number representation overflow. Since there was also a robot head returned, this was evaluated as correctly identifying ChatGPT-generated text, but the non-numerical percentage might confuse instructors using the tool.

The operation of a few of the tools was not immediately clear to some of the authors and the handling of results was sometimes not easy to document. For example, in PlagiarismCheck the AI-Detection button was not always presented on the screen and it would only show the last four tests done. Interestingly, Turnitin often returned high similarity values for ChatGPT-generated text, especially for program code or program output. This was distracting, as the similarity results were given first, the AI-detection could only be accessed by clicking on a number above the text "AI" that did not look clickable, but was, see Fig. 12.

Discussion

Detection tools for AI-generated text do fail, they are neither accurate nor reliable (all scored below 80% of accuracy and only 5 over 70%). In general, they have been found to diagnose human-written documents as AI-generated (false positives) and often diagnose AI-generated texts as human-written (false negatives). Our findings are consistent with previously published studies (Gao et al. 2022; Anderson et al. 2023; Elkhatat et al. 2023; Demers 2023; Gewirtz 2023; Krishna et al. 2023; Pegoraro et al. 2023; van Oijen 2023; Wang et al. 2023) and substantially differ from what some detection tools for AI-generated text claim (Compilatio 2023; Crossplag.com 2023; GoWinston.ai 2023; Zero GPT 2023). The detection tools present a main bias towards classifying the output as human-written rather than detecting AI-generated content. Overall, approximately 20% of AI-generated texts would likely be misattributed to humans.

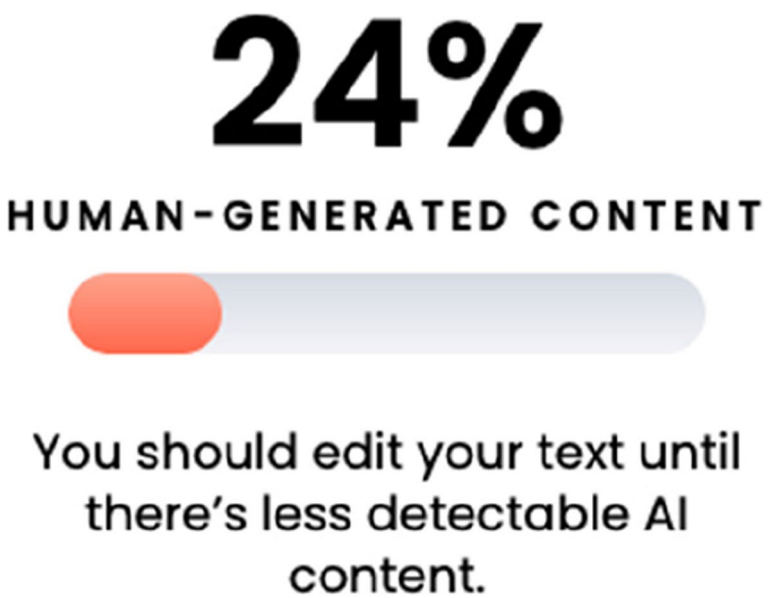


Fig. 13 Writer's suggestion to lower "detectable AI content"

They are neither robust, since their performance worsens even more with the use of obfuscation techniques such as manual editing or machine paraphrasing, nor are they able to cope with texts translated from other languages. Overall, approximately 50% of AI-generated texts that undergo some obfuscation would likely be misattributed to humans.

The results provided by the tools are not always easy to interpret for an average user. Some of them provide statistical information to justify the classification, and others highlight the text that is "likely" machine-generated. Some present values such as "perplexity = 137.222" or "Burstiness Score: 17104.959" with many digits of precision that do not generally help a user understand the results.

Some of the detection tools such as Writer are clearly aimed to be used to hide AI-written text, providing suggestions to users such as "You should edit your text until there's less detectable AI content." (See Fig. 13).

Detection tools for AI-generated text provide simple outputs with statements like "This document was likely written by AI" or "11% likely this comes from GPT-3, GPT-4 or ChatGPT", without any possibility of verification or evidence. Therefore, a student accused of unauthorised content generation only on this basis would have no possibility for a defence. The probability of false positives ranged from 0% (Turnitin) to 50% (GPT Zero). The probability of false negatives ranged from 8% (GPT Zero) to 100% (Content at Scale). The different types of failures may have serious implications. False positives could lead to wrong accusations of students, the false negatives allow students to evade detection of unauthorised content generation gaining unfair advantages and promoting impunity. Our experience and personal communications indicate that there is a large group of academics that believe in the output of the classifiers. The research results show that users should be extremely cautious when interpreting the results.

It is noteworthy that using machine translation such as Google translate or DeepL can lead to a higher number of false positives, leaving L2 students (and researchers) at risk of being falsely accused of unauthorised content generation when using machine translation to translate their own texts.

As the tools do not provide any evidence, the likelihood that an educational institution is able to prove this form of academic misconduct is extremely low. Reports provided by detection tools for AI-generated text cannot be used as the only basis for reporting students for cheating. They can give faculty a hint that some sort of misconduct may have happened, but further dialogue and conversations with students should take place.

One of the tools that the researchers came across, GLTR (http://gltr.io/) does not provide any classification, so it was decided to exclude it from testing. Nonetheless, it highlights the words (tokens) based on how commonly they appear in a given context. Interpretation of the output is up to the educator, but the research authors find the visualisation of this information very useful. The colour-coded predictability of individual words does not necessarily mean that the text was generated by AI, but may also mean that the text does not bring any innovation or added value, which might be—in some situations—a relevant indicator of its quality.

As the detection tools for AI-generated text are not reliable, a prevention-focused approach needs to be prioritised over a detection one. It is also paramount to inform the educators about this fact. The focus should instead be on the preventive

pedagogical strategies on how to ethically use generative AI tools, including a discussion about the benefits and limitations of such tools.

This presupposes defining, describing, and training on the differences between the ethical and unethical use of AI tools will be important for students, faculty, and staff. The ENAI recommendations on the ethical use of Artificial Intelligence in Education may be a good starting point (Foltýnek et al. 2023) for such discussions. It is also important to encourage educators to rethink their assessment strategies and instruments to achieve a design with features that reduce or even eliminate the possibility of enabling cheating.

Our study has some limitations. It focused only on English language texts. Even though we had computer code, we did not test the performance of the systems specifically on that. There were also indications that the results from the tools can vary when the same material is tested at a different time; we did not systematically examine the replicability of the results provided by the tools. Nevertheless, we tentatively suggest that this inconsistency can have major implications in misconduct investigations and thus provides another strong reason against the use of these tools as a single source of an accusation of misconduct. Our document set is also somewhat limited: we did not test the kind of hybrid writing with iterative use of AI that may be likely to be more typical of student use of generative AI. However, the poor performance of the tools across the range of documents does not imply better performance for hybrid writing.

Conclusion and future work

This paper exposes serious limitations of the state-of-the-art AI-generated text detection tools and their unsuitability for use as evidence of academic misconduct. Our findings do not confirm the claims presented by the systems. They too often present false positives and false negatives. Moreover, it is too easy to game the systems by using paraphrasing tools or machine translation. Therefore, our conclusion is that the systems we tested should not be used in academic settings. Although text matching software also suffers from false positives and false negatives (Foltýnek et al. 2020), at least it is possible to provide evidence of potential misconduct. In the case of the detection tools for AI-generated text, this is not the case.

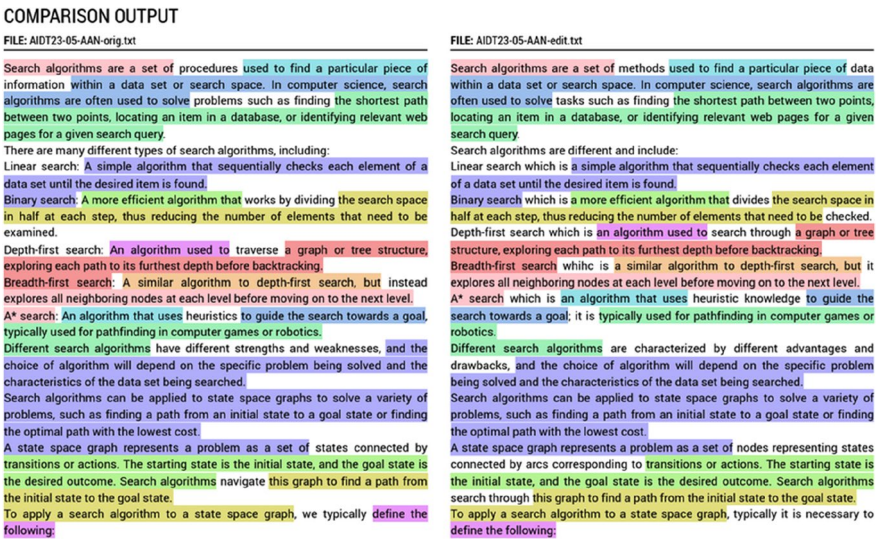
Our findings strongly suggest that the "easy solution" for detection of AI-generated text does not (and maybe even could not) exist. Therefore, rather than focusing on detection strategies, educators continue to need to focus on preventive measures and continue to rethink academic assessment strategies (see, for example, Bjelobaba 2020). Written assessment should focus on the process of development of student skills rather than the final product.

Future research in this area should test the performance of AI-generated text detection tools on texts produced with different (and multiple) levels of obfuscation e.g., the use of machine paraphrasers, translators, patch writers, etc. Another line of research might explore the detection of AI-generated text at a cohort level through its impact on student learning (e. g. through assessment scores) and education systems (e. g. the impact of generative AI on similarity scores). Research should also build on the known issues with cloud-based text-matching software to explore the legal implications and data privacy issues involved in uploading content to cloud-based (or institutional) AI detection tools.

Appendix

Case studies 05-ManEd

The following images show the generated texts on the left and the human-obfuscated ones on the right. The identical text is coloured in the same colour on both sides, with the changes popping out in white. The images were prepared using the *similarity-texter*. As can be seen, some texts were rather heavily re-written, others only had a few words exchanged.



COMPARISON OUTPUT

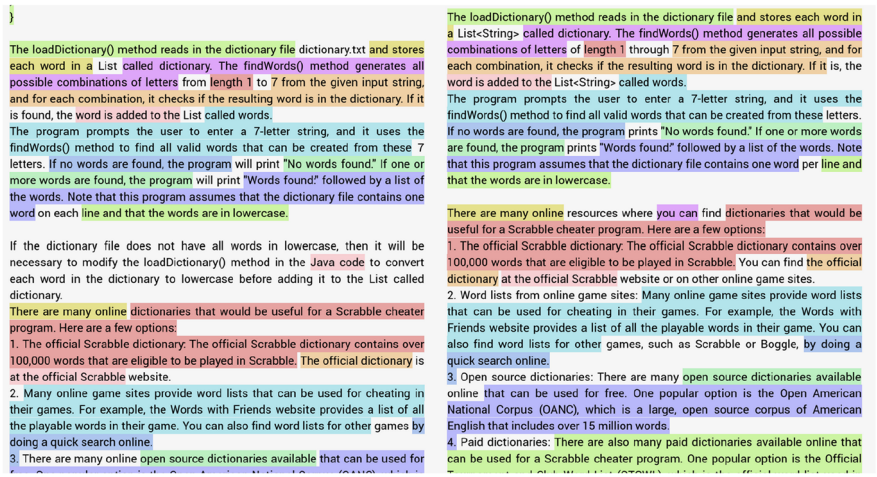
FILE: AIDT23-05-AAN-orig.txt

Search algorithms are a set of procedures used to find a particular piece of information within a data set or search space. In computer science, search algorithms are often used to solve problems such as finding the shortest path between two points, locating an item in a database, or identifying relevant web pages for a given search query.
There are many different types of search algorithms, including:
Linear search: A simple algorithm that sequentially checks each element of a data set until the desired item is found.
Binary search: A more efficient algorithm that works by dividing the search space in half at each step, thus reducing the number of elements that need to be examined.
Depth-first search: An algorithm used to traverse a graph or tree structure, exploring each path to its furthest depth before backtracking.
Breadth-first search: A similar algorithm to depth-first search, but instead explores all neighboring nodes at each level before moving on to the next level.
A* search: An algorithm that uses heuristics to guide the search towards a goal, typically used for pathfinding in computer games or robotics.
Different search algorithms have different strengths and weaknesses, and the choice of algorithm will depend on the specific problem being solved and the characteristics of the data set being searched.
Search algorithms can be applied to state space graphs to solve a variety of problems, such as finding a path from an initial state to a goal state or finding the optimal path with the lowest cost.
A state space graph represents a problem as a set of states connected by transitions or actions. The starting state is the initial state, and the goal state is the desired outcome. Search algorithms navigate this graph to find a path from the initial state to the goal state.
To apply a search algorithm to a state space graph, we typically define the following:

FILE: AIDT23-05-AAN-edit.txt

Search algorithms are a set of methods used to find a particular piece of data within a data set or search space. In computer science, search algorithms are often used to solve tasks such as finding the shortest path between two points, locating an item in a database, or identifying relevant web pages for a given search query.
Search algorithms are different and include:
Linear search which is a simple algorithm that sequentially checks each element of a data set until the desired item is found.
Binary search which is a more efficient algorithm that divides the search space in half at each step, thus reducing the number of elements that need to be checked.
Depth-first search which is an algorithm used to search through a graph or tree structure, exploring each path to its furthest depth before backtracking.
Breadth-first search whihc is a similar algorithm to depth-first search, but it explores all neighboring nodes at each level before moving on to the next level.
A* search which is an algorithm that uses heuristic knowledge to guide the search towards a goal; it is typically used for pathfinding in computer games or robotics.
Different search algorithms are characterized by different advantages and drawbacks, and the choice of algorithm will depend on the specific problem being solved and the characteristics of the data set being searched.
Search algorithms can be applied to state space graphs to solve a variety of problems, such as finding a path from an initial state to a goal state or finding the optimal path with the lowest cost.
A state space graph represents a problem as a set of nodes representing states connected by arcs corresponding to transitions or actions. The starting state is the initial state, and the goal state is the desired outcome. Search algorithms search through this graph to find a path from the initial state to the goal state.
To apply a search algorithm to a state space graph, typically it is necessary to define the following:

Fig. 14 AIDT23-05-AAN



}

The loadDictionary() method reads in the dictionary file dictionary.txt and stores each word in a List called dictionary. The findWords() method generates all possible combinations of letters from length 1 to 7 from the given input string, and for each combination, it checks if the resulting word is in the dictionary. If it is found, the word is added to the List called words.
The program prompts the user to enter a 7-letter string, and it uses the findWords() method to find all valid words that can be created from these 7 letters. If no words are found, the program will print "No words found." If one or more words are found, the program will print "Words found:" followed by a list of the words. Note that this program assumes that the dictionary file contains one word on each line and that the words are in lowercase.

If the dictionary file does not have all words in lowercase, then it will be necessary to modify the loadDictionary() method in the Java code to convert each word in the dictionary to lowercase before adding it to the List called dictionary.
There are many online dictionaries that would be useful for a Scrabble cheater program. Here are a few options:
1. The official Scrabble dictionary: The official Scrabble dictionary contains over 100,000 words that are eligible to be played in Scrabble. The official dictionary is at the official Scrabble website.
2. Many online game sites provide word lists that can be used for cheating in their games. For example, the Words with Friends website provides a list of all the playable words in their game. You can also find word lists for other games by doing a quick search online.
3. There are many online open source dictionaries available that can be used for

The loadDictionary() method reads in the dictionary file and stores each word in a List<String> called dictionary. The findWords() method generates all possible combinations of letters of length 1 through 7 from the given input string, and for each combination, it checks if the resulting word is in the dictionary. If it is, the word is added to the List<String> called words.
The program prompts the user to enter a 7-letter string, and it uses the findWords() method to find all valid words that can be created from these letters. If no words are found, the program prints "No words found." If one or more words are found, the program prints "Words found:" followed by a list of the words. Note that this program assumes that the dictionary file contains one word per line and that the words are in lowercase.

There are many online resources where you can find dictionaries that would be useful for a Scrabble cheater program. Here are a few options:
1. The official Scrabble dictionary: The official Scrabble dictionary contains over 100,000 words that are eligible to be played in Scrabble. You can find the official dictionary at the official Scrabble website or on other online game sites.
2. Word lists from online game sites: Many online game sites provide word lists that can be used for cheating in their games. For example, the Words with Friends website provides a list of all the playable words in their game. You can also find word lists for other games, such as Scrabble or Boggle, by doing a quick search online.
3. Open source dictionaries: There are many open source dictionaries available online that can be used for free. One popular option is the Open American National Corpus (OANC), which is a large, open source corpus of American English that includes over 15 million words.
4. Paid dictionaries: There are also many paid dictionaries available online that can be used for a Scrabble cheater program. One popular option is the Official

Fig. 15 AIDT23-05-DWW

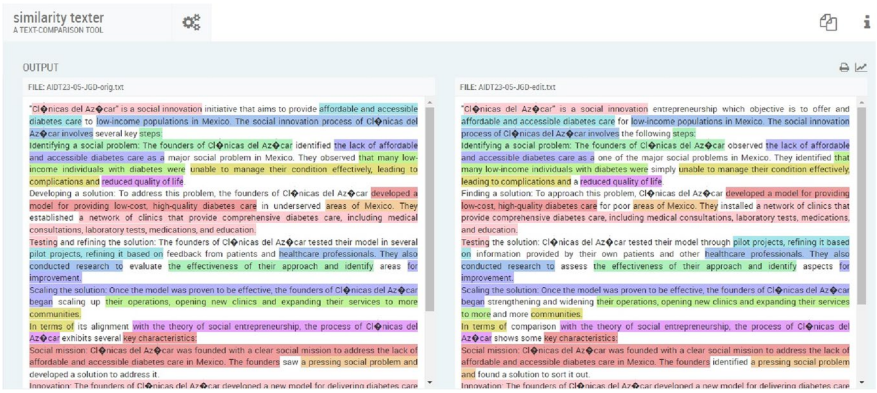


Fig. 16 AIDT23-05-JGD

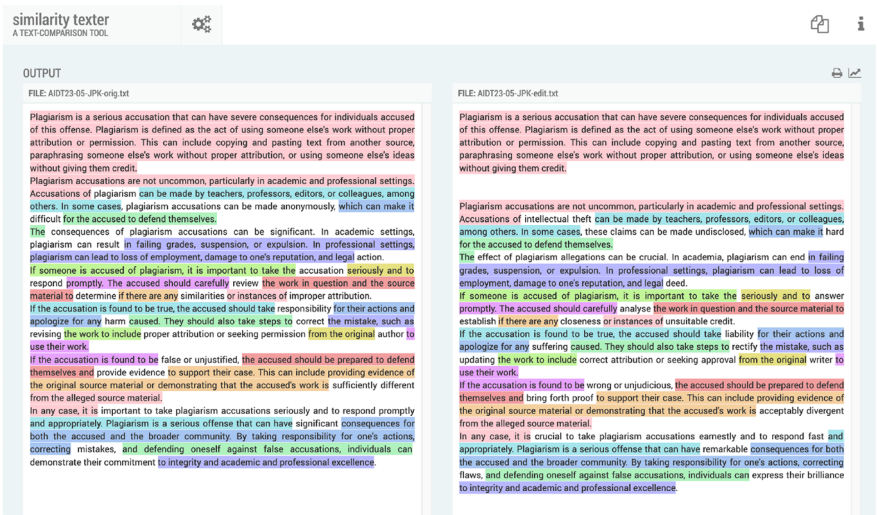


Fig. 17 AIDT23-05-JPK

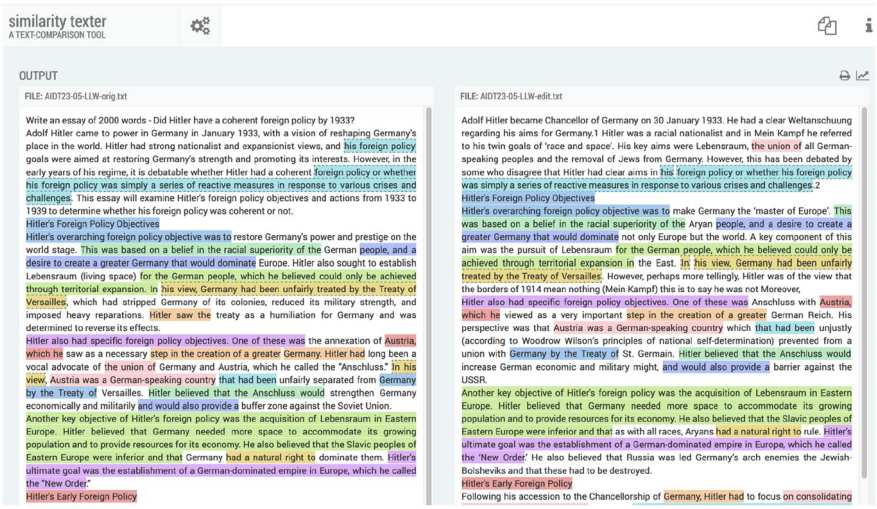


Fig. 18 AIDT23-05-LLW

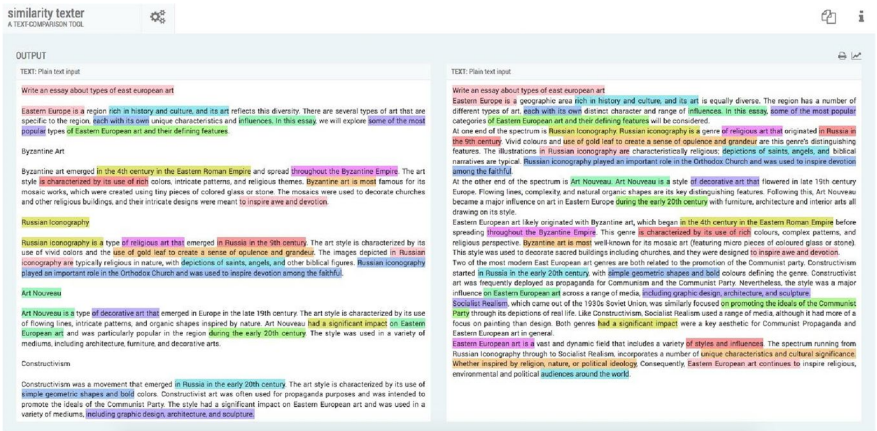


Fig. 19 AIDT23-05-OLU

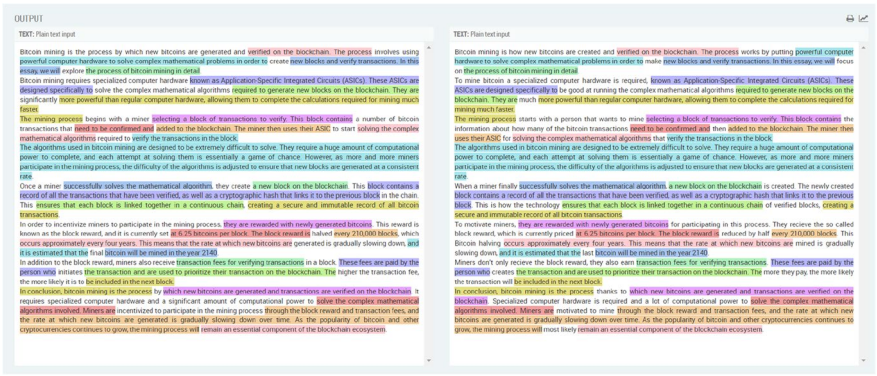


Fig. 20 AIDT23-05-PTR

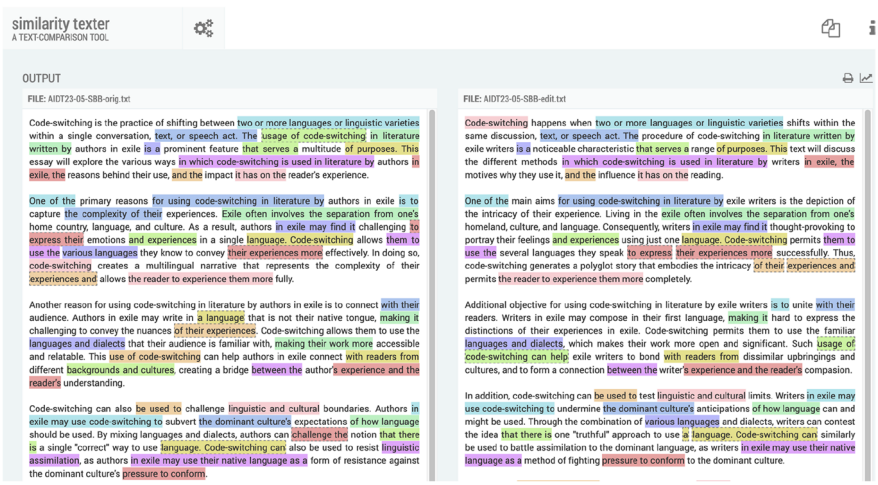
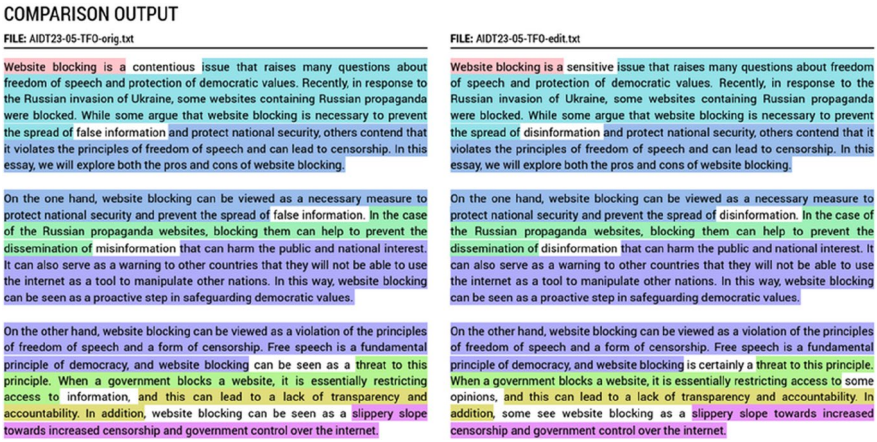


Fig. 21 AIDT23-05-SBB


COMPARISON OUTPUT

FILE: AIDT23-05-TFO-orig.txt

Website blocking is a contentious issue that raises many questions about freedom of speech and protection of democratic values. Recently, in response to the Russian invasion of Ukraine, some websites containing Russian propaganda were blocked. While some argue that website blocking is necessary to prevent the spread of false information and protect national security, others contend that it violates the principles of freedom of speech and can lead to censorship. In this essay, we will explore both the pros and cons of website blocking.

On the one hand, website blocking can be viewed as a necessary measure to protect national security and prevent the spread of false information. In the case of the Russian propaganda websites, blocking them can help to prevent the dissemination of misinformation that can harm the public and national interest. It can also serve as a warning to other countries that they will not be able to use the internet as a tool to manipulate other nations. In this way, website blocking can be seen as a proactive step in safeguarding democratic values.

On the other hand, website blocking can be viewed as a violation of the principles of freedom of speech and a form of censorship. Free speech is a fundamental principle of democracy, and website blocking can be seen as a threat to this principle. When a government blocks a website, it is essentially restricting access to information, and this can lead to a lack of transparency and accountability. In addition, website blocking can be seen as a slippery slope towards increased censorship and government control over the internet.

FILE: AIDT23-05-TFO-edit.txt

Website blocking is a sensitive issue that raises many questions about freedom of speech and protection of democratic values. Recently, in response to the Russian invasion of Ukraine, some websites containing Russian propaganda were blocked. While some argue that website blocking is necessary to prevent the spread of disinformation and protect national security, others contend that it violates the principles of freedom of speech and can lead to censorship. In this essay, we will explore both the pros and cons of website blocking.

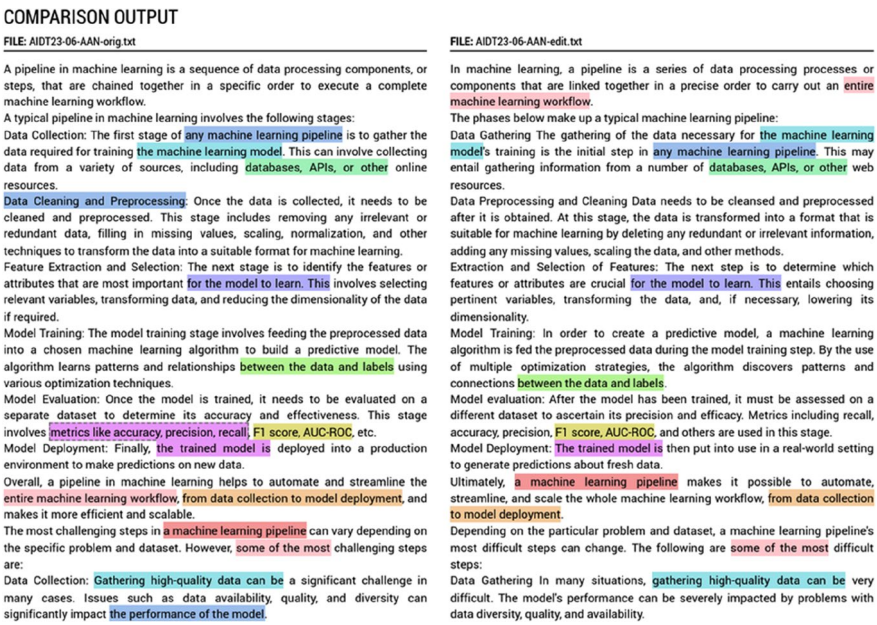
On the one hand, website blocking can be viewed as a necessary measure to protect national security and prevent the spread of disinformation. In the case of the Russian propaganda websites, blocking them can help to prevent the dissemination of disinformation that can harm the public and national interest. It can also serve as a warning to other countries that they will not be able to use the internet as a tool to manipulate other nations. In this way, website blocking can be seen as a proactive step in safeguarding democratic values.

On the other hand, website blocking can be viewed as a violation of the principles of freedom of speech and a form of censorship. Free speech is a fundamental principle of democracy, and website blocking is certainly a threat to this principle. When a government blocks a website, it is essentially restricting access to some opinions, and this can lead to a lack of transparency and accountability. In addition, some see website blocking as a slippery slope towards increased censorship and government control over the internet.

Fig. 22 AIDT23-05-TFO

Case studies 06-Para

These test cases were first generated with ChatGPT, then automatically re-written using Quillbot with the default settings. The generated original is on the left, the re-written version on the right.


COMPARISON OUTPUT

FILE: AIDT23-06-AAN-orig.txt

A pipeline in machine learning is a sequence of data processing components, or steps, that are chained together in a specific order to execute a complete machine learning workflow.

A typical pipeline in machine learning involves the following stages:

Data Collection: The first stage of any machine learning pipeline is to gather the data required for training the machine learning model. This can involve collecting data from a variety of sources, including databases, APIs, or other online resources.

Data Cleaning and Preprocessing: Once the data is collected, it needs to be cleaned and preprocessed. This stage includes removing any irrelevant or redundant data, filling in missing values, scaling, normalization, and other techniques to transform the data into a suitable format for machine learning.

Feature Extraction and Selection: The next stage is to identify the features or attributes that are most important for the model to learn. This involves selecting relevant variables, transforming data, and reducing the dimensionality of the data if required.

Model Training: The model training stage involves feeding the preprocessed data into a chosen machine learning algorithm to build a predictive model. The algorithm learns patterns and relationships between the data and labels using various optimization techniques.

Model Evaluation: Once the model is trained, it needs to be evaluated on a separate dataset to determine its accuracy and effectiveness. This stage involves metrics like accuracy, precision, recall, F1 score, AUC-ROC, etc.

Model Deployment: Finally, the trained model is deployed into a production environment to make predictions on new data.

Overall, a pipeline in machine learning helps to automate and streamline the entire machine learning workflow, from data collection to model deployment, and makes it more efficient and scalable.

The most challenging steps in a machine learning pipeline can vary depending on the specific problem and dataset. However, some of the most challenging steps are:

Data Collection: Gathering high-quality data can be a significant challenge in many cases. Issues such as data availability, quality, and diversity can significantly impact the performance of the model.

FILE: AIDT23-06-AAN-edit.txt

In machine learning, a pipeline is a series of data processing processes or components that are linked together in a precise order to carry out an entire machine learning workflow.

The phases below make up a typical machine learning pipeline:

Data Gathering The gathering of the data necessary for the machine learning model's training is the initial step in any machine learning pipeline. This may entail gathering information from a number of databases, APIs, or other web resources.

Data Preprocessing and Cleaning Data needs to be cleansed and preprocessed after it is obtained. At this stage, the data is transformed into a format that is suitable for machine learning by deleting any redundant or irrelevant information, adding any missing values, scaling the data, and other methods.

Extraction and Selection of Features: The next step is to determine which features or attributes are crucial for the model to learn. This entails choosing pertinent variables, transforming the data, and, if necessary, lowering its dimensionality.

Model Training: In order to create a predictive model, a machine learning algorithm is fed the preprocessed data during the model training step. By the use of multiple optimization strategies, the algorithm discovers patterns and connections between the data and labels.

Model evaluation: After the model has been trained, it must be assessed on a different dataset to ascertain its precision and efficacy. Metrics including recall, accuracy, precision, F1 score, AUC-ROC, and others are used in this stage.

Model Deployment: The trained model is then put into use in a real-world setting to generate predictions about fresh data.

Ultimately, a machine learning pipeline makes it possible to automate, streamline, and scale the whole machine learning workflow, from data collection to model deployment.

Depending on the particular problem and dataset, a machine learning pipeline's most difficult steps can change. The following are some of the most difficult steps:

Data Gathering In many situations, gathering high-quality data can be very difficult. The model's performance can be severely impacted by problems with data diversity, quality, and availability.

Fig. 23 AIDT23-06-AAN

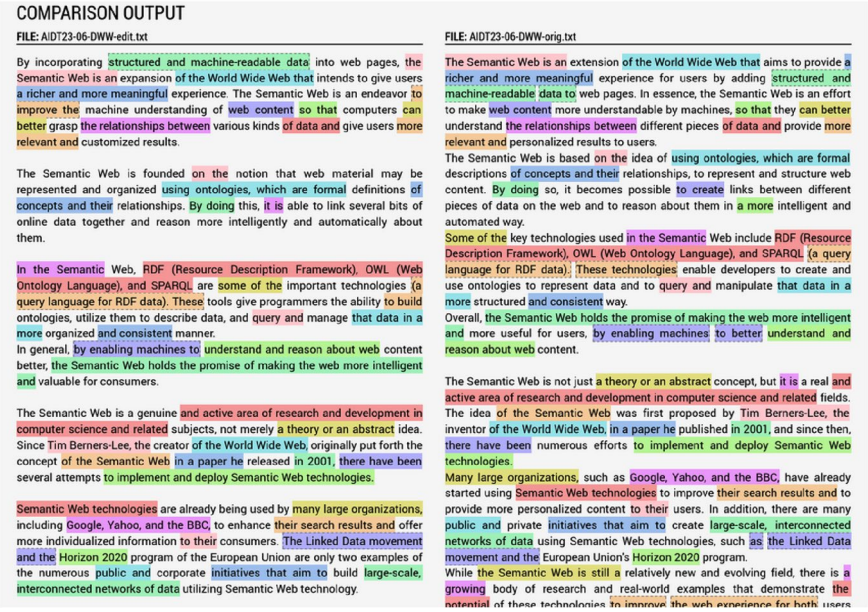


Fig. 24 AIDT23-06-DWW

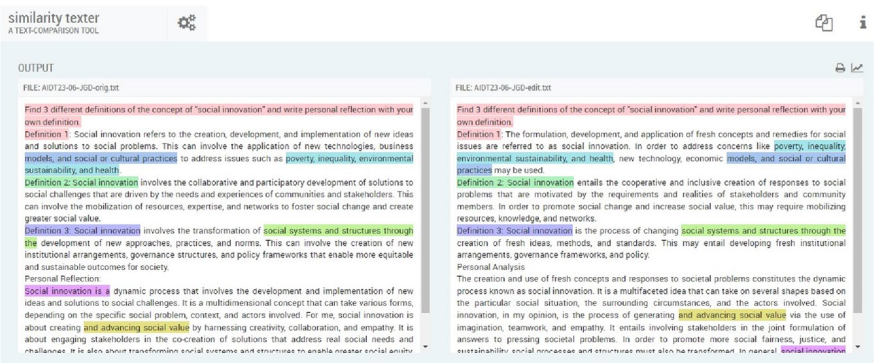


Fig. 25 AIDT23-06-JGD

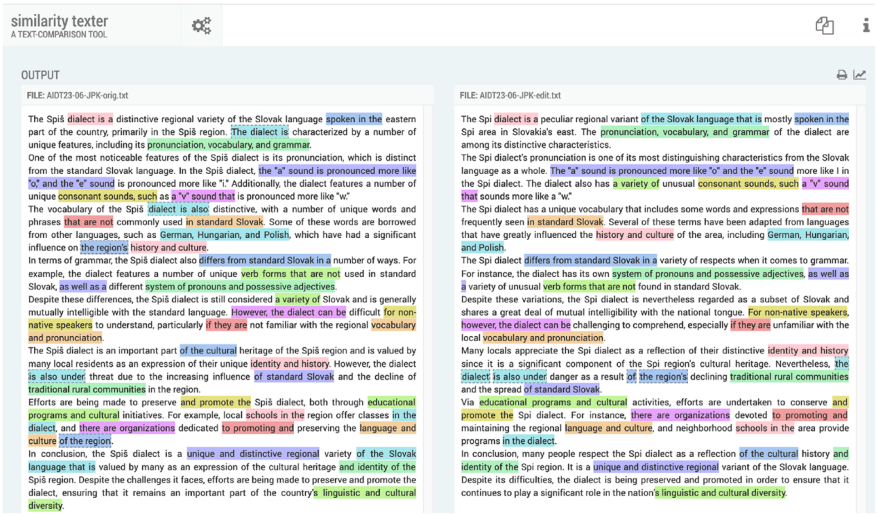


Fig. 26 AIDT23-06-JPK

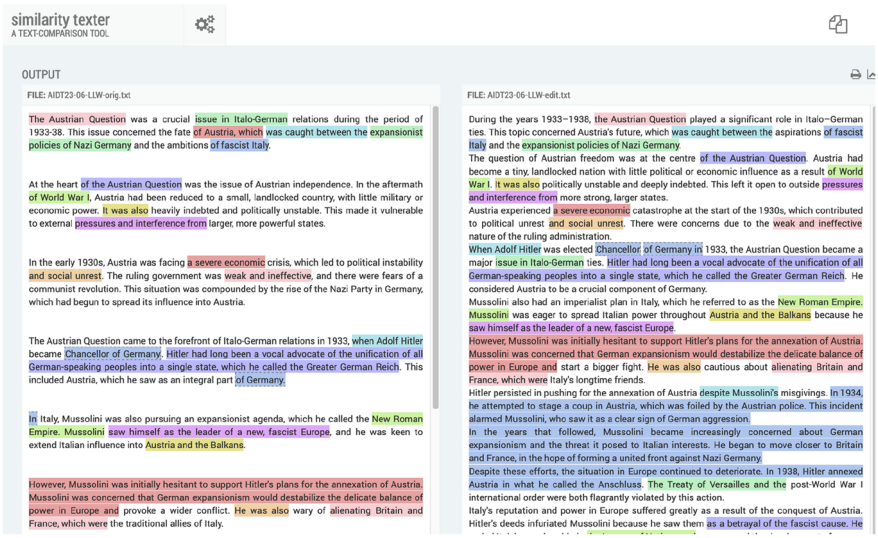


Fig. 27 AIDT23-06-LLW

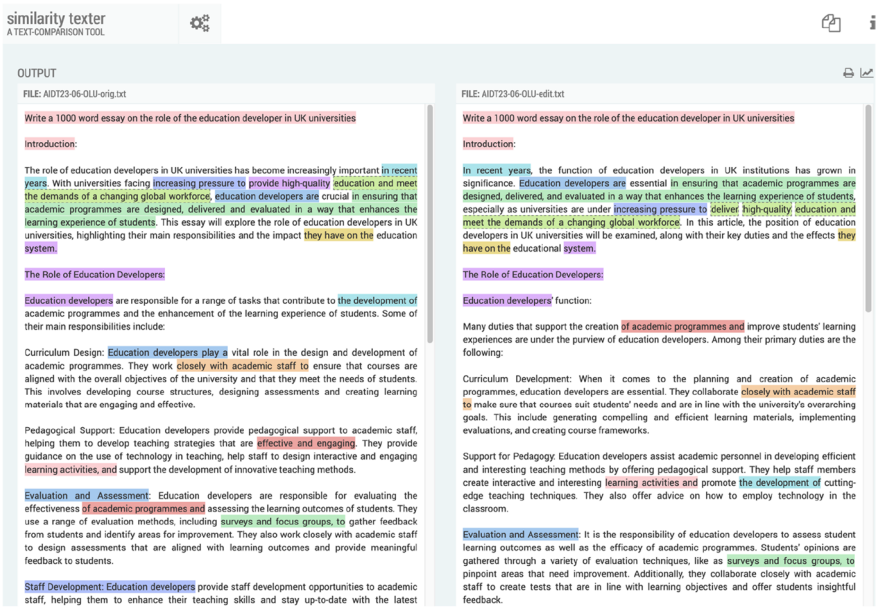


Fig. 28 AIDT23-06-OLU

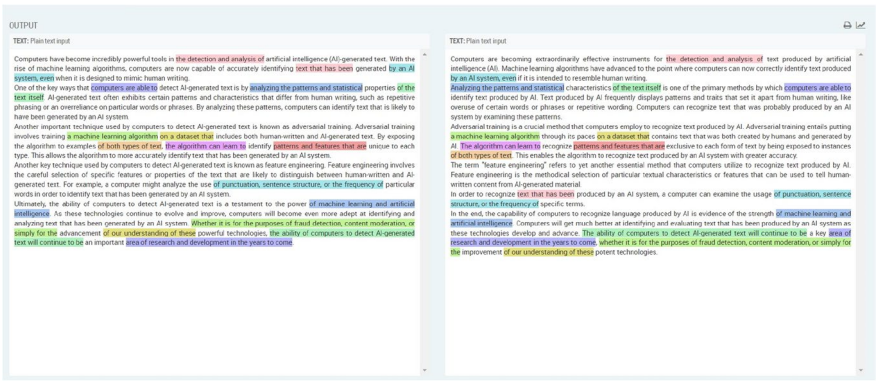


Fig. 29 AIDT23-06-PTR

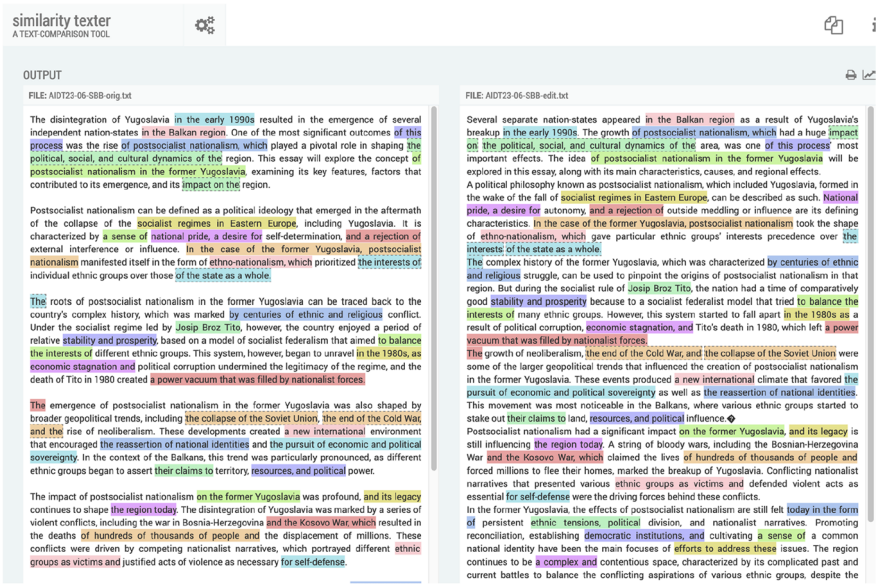


Fig. 30 AIDT23-06-SBB

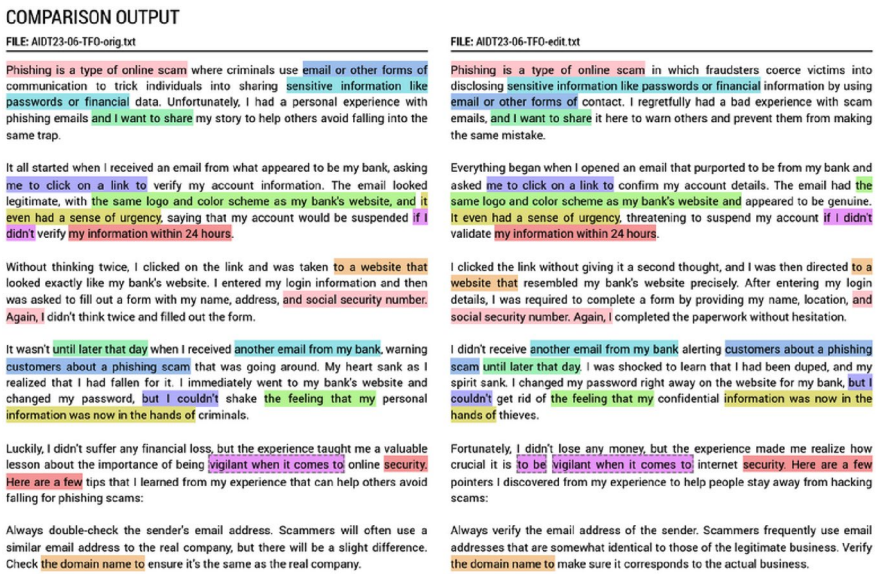


Fig. 31 AIDT23-06-TFO

Abbreviations

01-Hum	Human-written
02-MT	Human-written in a non-English language with a subsequent AI/machine translation to English
03-AI	AI-generated text
04-AI	AI-generated text with subsequent human manual edits
05-ManEd	AI-generated text with subsequent manual paraphrase by human

06-Para	AI-generated text with subsequent AI/machine paraphrase
ACC	Accuracy
ACC_bin	Accuracy, binary approach
ACC_SEMIBIN	Accuracy, semi-binary approach
AI	Artificial intelligence
GPT	Generative pre-trained transformer
FAS	False accusation
FN	False negative
FP	False positive
HEIs	Higher education institutions
LLM	Large language models
NaN	Not a number
PFN	Partially false negative
PFP	Partially false positive
PTP	Partially true positive
PTN	Partially true negative
TN	True negative
TP	True positive
UNC	Unclear

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s40979-023-00146-z.

Additional file 1. Supplementary material: Raw data.

Acknowledgements
The authors wish to thank their colleague Július Kravjar from Slovakia who contributed a full set of test documents to the investigation.
The authors also wish to thank their colleagues from Turkey, Salim Razı and Özgür Çelik, who participated in the initial stages of the discussions about this research endeavour, but due to the devastating earthquake in February 2023 were not able to contribute further.
The tool *similarity-texter* was created as part of the bachelor's thesis of Sofia Kalaidopoulou and is based on Dick Grune's sim_text algorithm. It was submitted to the HTW Berlin in 2016 and is available under a Creative Commons BY-NC-SA 4.0 International License at https://people.f4.htw-berlin.de/~weberwu/simtexter/app.html.
ChatGPT was NOT used to tweak any portion of this publication.

Authors' contributions
All authors created test data, ran the tests, collected data, discussed the statistical results, and contributed equally to the text. TF and OP prepared the statistics for discussion.

Authors' information
The authors are members of the European Network for Academic Integrity (ENAI) working group on Technology and Academic Integrity. DWW is a plagiarism researcher and a retired professor of computer science from the HTW Berlin, Germany. AAN is an associate professor at the Department of Artificial Intelligence and Systems Engineering of Riga Technical University, Latvia. SB is a researcher in research integrity at Center for Research Ethics & Bioethics, at Uppsala University, Sweden, and the Vice-president of ENAI. TF is an assistant professor at the Department of Machine Learning and Data Processing at the Faculty of Informatics, Masaryk University, Czechia, and President of ENAI. JGD is a professor of the School of Engineering from University of Monterrey, Mexico and oversees the efforts of its Center for Integrity and Ethics. OP is an Education Developer specialising in assessment integrity at Queen Mary University of London, UK. PS is a student of Computer Science at the Faculty of Informatics, Masaryk University, Czechia. LW is the University of Leeds, UK, Academic Integrity Lead.

Funding
Open access funding provided by Uppsala University. The authors had no funding for this research other than from their respective institutions.

Availability of data and materials
All data and testing materials are available at https://www.academicintegrity.eu/wp/technology-academic-integrity-working-group/.

Declarations

Competing interests
Two authors of this article, SB and TF, are involved in organising the European Conference on Ethics and Integrity in Academia 2023, co-organised by the European Network for Academic Integrity. This conference receives sponsorship from Turnitin and Compilatio. This did not influence the research presented in the paper in any phase.
Three of the authors, JGD, SB and TF are members of the editorial board of the ***International Journal for Educational Integrity***. They can thus not act as reviewers.
One author, TF, is guest editor for the special issue on Artificial Intelligence.

Received: 19 July 2023 Accepted: 19 October 2023
Published online: 25 December 2023

References

Anderson N, Belavy DL, Perle SM, Hendricks S, Hespanhol L, Verhagen E, Memon AR (2023) AI did not write this manuscript, or did it? Can we trick the AI text detector into generated texts? The potential future of ChatGPT and AI in Sports & Exercise Medicine manuscript generation. BMJ Open Sport Exerc Med 9:e001568. https://doi.org/10.1136/bmjsem-2023-001568

Aydın Ö, Karaarslan E (2022) OpenAI ChatGPT Generated Literature Review: Digital Twin in Healthcare. In: Aydın Ö (ed) Emerging Computer Technologies 2. İzmir Akademi Dernegi, pp 22–31

Bender EM, Gebru T, McMillan-Major A, Shmitchell S (2021) On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. ACM, New York, pp 610–623. https://doi.org/10.1145/3442188.3445922

Bjelobaba S (2020) Academic Integrity Teacher Training: Preventive Pedagogical Practices on the Course Level. In: Khan Z, Hill C, Foltýnek T (eds) Integrity in Education for Future Happiness. Mendel University in Brno, Brno, pp 9–18 (http://academicintegrity.eu/conference/proceedings/2020/bjelobaba.pdf)

Borji A. (2023). A Categorical Archive of ChatGPT Failures. arXiv. https://doi.org/10.48550/arXiv.2302.03494

Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Amodei D. (2020). Language Models are Few-Shot Learners. arXiv. https://doi.org/10.48550/arXiv.2005.14165

Chakraborty S, Bedi AS, Zhu S, An B, Manocha D, Huang F. (2023) On the Possibilities of AI-Generated Text Detection. arXiv. https://doi.org/10.48550/arXiv.2304.04736

Clarke R, Lancaster T. (2006). Eliminating the successor to plagiarism? Identifying the usage of contract cheating sites. Proceedings of 2nd International Plagiarism Conference Newcastle, UK, 14

Compilatio (2023). Comparison of the best AI detectors in 2023 (ChatGPT, YouChat...). https://www.compilatio.net/en/blog/best-ai-detectors. Accessed 12 April 2023

Content at Scale (2023). How accurate is this for AI detection purposes? https://contentatscale.ai/ai-content-detector/. Accessed 8 May 2023

Crossplag.com (2023). How accurate is the AI Detector? https://crossplag.com/ai-content-detector/. Accessed 8 May 2023

Demers T. (2023). 16 of the best AI and ChatGPT content detectors compared. Search Engine Land. https://searchengineland.com/ai-chatgpt-content-detectors-395957. Accessed May 9 2023

Devlin J, Chang MW, Lee K, Toutanova K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (pp. 4171–4186)x. Minneapolis, Minnesota. Association for Computational Linguistics

Elkhatat AM, Elsaid K, Almeer S (2023) Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. Int J Educ Integrity 19:17. https://doi.org/10.1007/s40979-023-00140-5. (19(1), 1-16)

Elsen-Rooney M. (2023). NYC education department blocks ChatGPT on school devices, networks. Chalkbeat New York. https://ny.chalkbeat.org/2023/1/3/23537987/nyc-schools-ban-chatgpt-writing-artificial-intelligence. Accessed 14 June 2023

Foltýnek T, Dlabolová D, Anohina-Naumeca A, Razı S, Kravjar J, Kamzola L, Guerrero-Dib J, Çelik Ö, Weber-Wulff D (2020) Testing of support tools for plagiarism detection. Int J Educ Technol High Educ 17(1):1–31. https://doi.org/10.1186/s41239-020-00192-4

Foltýnek T, Bjelobaba S, Glendinning I, Khan ZR, Santos R, Pavletic P, Kravjar J (2023) ENAI Recommendations on the ethical use of Artificial Intelligence in Education. Int J Educ Integrity 19(1):1. https://doi.org/10.1007/s40979-023-00133-4

Gao CA, Howard FM, Markov NS, Dyer EC, Ramesh S, Luo Y, Pearson AT. (2022) Comparing scientific abstracts generated by ChatGPT to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. bioRxiv. https://doi.org/10.1101/2022.12.23.521610

Gewirtz D. (2023). Can AI detectors save us from ChatGPT? I tried 3 online tools to find out. https://www.zdnet.com/article/can-ai-detectors-save-us-from-chatgpt-i-tried-3-online-tools-to-find-out/. Accessed 8 May 2023

GoWinston.ai. (2023). "Are AI detection tools accurate?" Winston AI | The most powerful AI content detector. https://gowinston.ai/. Accessed 8 May 2023

GPTZero. (2023). The Global Standard for AI Detection:Humans Deserve the Truth. https://gptzero.me/. Accessed 8 May 2023

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. arXiv. https://doi.org/10.48550/arXiv.2301.07597

Howard RM (1995) Plagiarisms, Authorships, and the Academic Death Penalty. Coll Engl 57(7):788–806. https://doi.org/10.2307/378403

ICML. (2023). ICML 2023 Call For Papers, Fortieth International Conference on Machine Learning. https://icml.cc/Conferences/2023/CallForPapers. Accessed 14 June 2023

Ippolito D, Duckworth D, Callison-Burch C, Eck D. (2020). Automatic Detection of Generated Text is Easiest when Humans are Fooled. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp.1808–1822). https://doi.org/10.18653/v1/2020.acl-main.164/

Johnson A. (2023). ChatGPT In Schools: Here's Where It's Banned—And How It Could Potentially Help Students. Forbes. https://www.forbes.com/sites/ariannajohnson/2023/01/18/chatgpt-in-schools-heres-where-its-banned-and-how-it-could-potentially-help-students/. Accessed 14 June 2023

Khalil M, Er E. (2023). Will ChatGPT get you caught? Rethinking of Plagiarism Detection. EdArXiv. https://doi.org/10.35542/osf.io/fnh48
Krishna K, Song Y, Karpinska M, Wieting J, Iyyer M. (2023). Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. arXiv. https://doi.org/10.48550/arXiv.2303.13408
Liyanage V, Buscaldi D, Nazarenko A. (2022). A Benchmark Corpus for the Detection of Automatically generated Text in Academic Publications. Proceedings of the 13th Conference on Language Resources and Evaluation (pp. 4692–4700). European Language Resources Association
Ma Y, Liu J, Yi F, Cheng Q, Huang Y, Lu W, Liu X. (2023). AI vs. Human - Differentiation Analysis of Scientific Content Generation. arXiv. https://doi.org/10.48550/arXiv.2301.10416
Marr B. (2023). A Short History Of ChatGPT: How We Got To Where We Are Today. Forbes. https://www.forbes.com/sites/bernardmarr/2023/05/19/a-short-history-of-chatgpt-how-we-got-to-where-we-are-today/. Accessed 14 June 2023
Mikolov T, Chen K, Corrado G, Dean J. (2013). Efficient estimation of word representations in vector space. arXiv. https://doi.org/10.48550/arXiv.1301.3781
Milmo D. (2023). ChatGPT reaches 100 million users two months after launch. The Guardian. https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app. Accessed 14 June 2023
van Oijen V. (2023). AI-generated text detectors: Do they work? SURF Communities. https://communities.surf.nl/en/ai-in-education/article/ai-generated-text-detectors-do-they-work. Accessed 8 May 2023
OpenAI. (2023). ChatGPT February 13 Version. https://chat.openai.com/
OpenAI. (2023). New AI classifier for indicating AI-written text. https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text
Pegoraro A, Kumari K, Fereidooni H, Sadeghi AR. (2023). To ChatGPT, or not to ChatGPT: That is the question! arXiv. https://doi.org/10.48550/arXiv.2304.01487
Quillbot (2023). Quillbot AI Paraphrasing Tool. https://quillbot.com/
Rosenfeld R (2000) Two decades of statistical language modeling: Where do we go from here? Proc IEEE 88(8):1270–1278. https://doi.org/10.1109/5.880083t
Schechner S. (2023). ChatGPT Ban Lifted in Italy After Data-Privacy Concessions. Wall Street J. https://www.wsj.com/articles/chatgpt-ban-lifted-in-italy-after-data-privacy-concessions-d03d53e7. Accessed 14 June 2023
Tauginienė L, Gaižauskaitė I, Glendinning I, Kravjar J, Ojstršek M, Ribeiro L, Odineca T, Marino F, Cosentino M, Sivasubramaniam S. (2018). Glossary for Academic Integrity. ENAI. http://www.academicintegrity.eu/wp/wp-content/uploads/2018/02/GLOSSARY_final.pdf. Accessed 14 June 2023
Turnitin (2023). Understanding false positives within our AI writing detection capabilities. https://www.turnitin.com/blog/understanding-false-positives-within-our-ai-writing-detection-capabilities. Accessed 14 June 2023
Turnitin (2023). Resources to Address False Positives.Turnitin Support. https://supportcenter.turnitin.com/s/article/Turnitin-s-AI-Writing-Detection-Toolkit-for-administrators-and-instructors. Accessed 8 May 2023
Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. (2017). Attention is all you need. Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Advances in Neural Information Processing systems, USA. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. Accessed 8 May 2023
Wang J, Liu S, Xie X, Li Y. (2023). Evaluating AIGC Detectors on Code Content. arXiv. https://doi.org/10.48550/arXiv.2304.05193
Zero GPT (2023). What is the accuracy rate of ZeroGPT? ZeroGPT - Chat GPT, Open AI and AI text detector Free Too. https://www.zerogpt.com/. Accessed 8 May 2023

Publisher's Note