Automation of Plagiarism and AI Detection in IT Students Papers with a Software Tool Using API

Oleksandr Sharyhin[1,2,3], Oksana Klochko[1,6] and Ihor Tverdokhlib[2,4,5]
[1] *Vinnytsia Mykhailo Kotsiubynskyi State Pedagogical University, Ostrozhskogo Str. 32, 21100 Vinnytsia, Ukraine*
[2] *Dragomanov Ukrainian State University, Pyrogova Str. 9, 01601 Kyiv, Ukraine*
[3] *Miratech, Vatslav Havel Avenue 6z, 03124 Kyiv, Ukraine*
[4] *Institute of Pedagogy of the NAES of Ukraine, Sichovyh Streltsiv Str. 52-D, 04053 Kyiv, Ukraine*
[5] *Taras Shevchenko National University of Kyiv, Volodymyrska Str. 60, 01033 Kyiv, Ukraine*
[6] *Academy of cognitive and Natural Sciences, 54 Universytetskyi Ave., Kryvyi Rih, 50086, Ukraine*
{exhaustic, klochkoob, i.a.tverdokhlib}@gmail.com

Keywords: Academic Integrity, Plagiarism, Automation, Student Work Verification, Information Technology Students, Professional Training, Artificial Intelligence, Information Law, Natural Language Processing, Intelligent Information System.

Abstract: The article considers various approaches to automating the checking of student works by lecturers for plagiarism and the presence of fragments of text/code generated by Artificial Intelligence. As a result of testing various text fragments related to IT, it was concluded that some of existing services can be considered as good candidates for plagiarism detection, but there's no obvious choice for AI detection. Services (or combination of them) may act as a support mechanism, but final decisions should be made by lecturers. A software application has been developed to automate the routine work of lecturers of higher education institutions in checking student works for plagiarism and AI detection. Developed software application uses various APIs to search for plagiarism and fragments of text/code generated by AI. The practical value of the study lies in the possibility of using the proposed method to check student works for originality. Further research on this topic may include multi–user support, integration with existing educational platforms, adding UI components for visualization of AI/plagiarism detection results.

1 INTRODUCTION

The development of information technologies and their mass introduction into most spheres of human activity, in particular into the educational process, has not only positive aspects. Currently, the Internet contains a large amount of information, the amount of which is constantly growing. Systems based on Artificial Intelligence are actively developed and used, which are capable of generating almost any information content. All this creates the prerequisites for unscrupulous researchers who present materials found on the Internet or generated by generative Artificial Intelligence (AI) as their own scientific results.

In the conditions of the Information Society, the training of future specialists in information technologies plays an important role. They are the driver of the further development of the sphere of information technologies, playing a key role in most areas of human activity: economy, production, medicine, service sector, education, etc. However, due to the conditions defined above, the training of future specialists in information technologies is accompanied by the conscious or accidental use of borrowings in writing scientific papers, program code, borrowing other people's ideas. Therefore, the problem of detecting plagiarism in students works is a very important and urgent task of today, and the solution to this task usually falls on the shoulders of lecturers of higher education institutions.

A review of scientific sources on solving the problem of anti-plagiarism verification of scientific and current works of students was conducted. According to the results of the analysis, the main directions of scientific research of scientists were identified. For more than ten years, the problem of detecting plagiarism and developing programs for anti-plagiarism verification has been put to the fore. In scientific works devoted to the study of these issues, the authors raise the issues of detecting plagiarism in scientific and pedagogical activities [1, 2], the use of AI systems by students for writing papers [3, 4], and a review and comparative analysis of existing programs for anti-plagiarism verification.

An important role in observing the principles of academic integrity in the training of future IT specialists is played by the detection of plagiarism in the program code during the lecturer's verification of student works or educational projects. In the article [5] the authors present a privacy-preserving protocol for plagiarism detection that eliminates the need for code disclosure during similarity computation. In the works [6, 7] scientific investigations on ways to automate the verification of the source code in student scientific works were published. A group of developers in [8] developed a web-based application that contains a set of tools for detecting and preventing plagiarism in educational program code.

The active use of AI systems in all spheres of human activity [9] in recent years contributes to an increase in the generation of borrowings by students in their scientific papers and program code. On the other hand, the use of AI methods and systems contributes to the automation of plagiarism detection. Developments have also been underway in this direction in recent years. In particular, it is worth highlighting a number of works [10, 11] that consider methods for detecting plagiarism based on semantic text analysis and deep learning algorithms. In work [12], the authors proposed a fundamentally new approach for the estimation of the time complexity of an algorithm. However, the issues of automating the verification of student scientific and practical papers using several anti-plagiarism verification services and combining different types of services (detecting matches with information sources on the Internet, using artificial intelligence systems to analyze texts for borrowings or fabrication) to check the work for plagiarism remain insufficiently developed.

The authors provide a list of processes that can be automated [13, 14] and a comparative analysis of ready-made software products for automated testing.

2 LITERATURE REVIEW

According to the analysis of scientific sources, popular science publications and normative documents, we have identified the main concepts that form the methodological basis of the research.

On the basis of a review of normative documents and scientific sources, the authors define the definition of plagiarism as the use of a part of other people's public works or scientific papers, texts, images, ideas without attribution in order to present them as one's own research results. Detection of plagiarism plays an important role in the educational sector, especially when it comes to the presentation of research results by students.

There are various classifications of plagiarism types. Summarizing the results presented in [15, 16], the following main types of plagiarism can be distinguished:

- Complete Plagiarism is a type of plagiarism when you present a work written by another person as your own.
- Direct Plagiarism – copying individual parts of someone else's work (words, sentences, paragraphs or chapters) and adding them to your work without citing the author.
- Paraphrasing Plagiarism is one of the most common types of plagiarism, which involves borrowing someone else's idea, paraphrasing it in your own words and placing it in your work without attribution.
- Plagiarizing Yourself – reusing your own ideas and parts of texts that have already been published.
- Accidental Plagiarism is a type of plagiarism according to which parts of other works are unintentionally used in writing a part of a work without giving a reference to them. This is usually done unconsciously or accidentally.
- Mosaic Plagiarism is the use by an author of very small parts of the works of other authors in combination with his or her own thoughts.
- Source-based Plagiarism is the use of borrowings together with references to their source. In this case, the source may be incorrect, not be the original source, or not exist.

In recent years, revolutionary processes have been observed related to the active distribution and use of artificial intelligence-based technologies in various fields of human activity. Artificial Intelligence is also actively penetrating into the educational sphere. Students of higher education institutions are increasingly using and abusing AI-based tools. In [17], it is noted that "...students put the morality of the act in whether they manipulate the tool or not, and not so much in the intellectual property of what is delivered". Since AI is currently a tool that has no moral principles and cannot independently ensure the principles of academic integrity, all responsibility for the use of AI tools lies with its users [17].

The World Economic Forum notes that our society is navigating a second era of digital technologies, which include publicly available generative artificial intelligence [18]. Such systems allow students to generate texts for written work quite easily. Thus, developments in the field of AI have benefited many industries, but for the educational and scientific field, they have created new challenges for students to use AI tools to generate scientific and educational texts, they have reinforced the problem of academic integrity in the modern information society.

Today there are international organizations to combat cheating, plagiarism, and academic dishonesty in higher education. Such an organization is The International Center for Academic Integrity [19]. It offers assessment services, resources, and consultations to its member institutions. The European Network of Academic Integrity (ENAI) holds an annual conference on ethics and integrity in academia. At the last conference significant debate was generated by the session on artificial intelligence (AI) detection tools based on recent research by a team of ENAI members concluding that they don't work reliably [20].

The European Commission issued the first act on the regulation of AI [21]. It is the first comprehensive legal framework for artificial intelligence worldwide. It regulates the rules for the use of AI in education and professional training. Content that is either generated or modified with the help of AI — images, audio or video files (for example, deepfakes) — needs to be clearly labeled as AI-generated so that users are aware when they come across such content. Thus, it is planned to regulate the issue of academic integrity regarding the use of AI systems.

Plagiarism detection software was developed in the 1990s. These programs were focused on detecting plagiarism in the form of copying and pasting data from the Internet [2]. However, this problem has now become much more complicated due to the ability of artificial intelligence systems to generate texts and program code. The search strategies used to detect fabrications made by artificial intelligence differ from those used to detect plagiarism [3]. Plagiarism detection is a part of natural language processing (NLP). Currently, there are many solutions for detecting lexical or syntactic plagiarism based on NLP methods, in particular the concept of using language dictionaries such as WordNet [22].

Nowadays, scientists have made a significant contribution to solving the problem of automated verification of students' academic texts, but the sources and types of plagiarism are constantly changing and new ones are emerging. Therefore, the problem of automating the detection of plagiarism in students' academic papers is relevant and will remain so in the future.

The purpose of the study is to automate the routine work of higher education lecturers in checking IT student papers for plagiarism and AI detection by developing a software application using the API to search for plagiarism and fragments of text/code generated by AI.

3 METHODOLOGY

The research was conducted through an empirical analysis of various API-based services to detect plagiarism and AI-generated content in academic texts. All text fragments that we used in this research are related to IT. The methodology involved the creation and classification of text datasets and the systematic testing of external services using these text datasets.

For the purposes of this study, two distinct datasets were compiled: one is for evaluating AI-generated content detection, and another designed to assess the accuracy of plagiarism detection. Three categories of text fragments were prepared for AI detection:

1) Data subset DS11. AI-generated texts – generated using previous versions of ChatGPT (GPT-3.5 and earlier). Our assumption was that existing services would handle AI detection for these cases, showing results close to
   a) 80 text fragments were generated.
2) Data subset DS12. AI-generated texts – generated using latest version of ChatGPT (GPT-4o). This service is newer and is available in paid plans. Our assumption was that existing services would handle AI detection for these cases, showing results close to 100. 60 text fragments were generated.
3) Data subset DS13. Original academic texts
   a) from dissertations written before the emergence of generative AI tools i.e. there is not even a theoretical possibility that text fragments were generated with the help of AI. 112 text fragments were prepared.

For plagiarism detection, a separate dataset was constructed, it contains three categories of text fragments:

1) Data subset DS21. Existing scientific articles, dissertation works. We expect to find plagiarism in these documents because this information already exists and it's public. 134 text fragments were prepared.

2) Data subset DS22. Fragments of new texts which were not published yet. We expect not to find plagiarism in these fragments. 56 text fragments were prepared.

3) Data subset DS23. Text fragments generated by different versions of AI text generators. We expect not to find plagiarism in these fragments. 31 text fragments were generated.

The analysis of API service outputs on various categories of text fragments reveals the occurrence of both Type I errors (false positives) and Type II errors (false negatives).

To prove the statistical significance of the results and justify that the selected number of fragments is sufficient for reliable conclusions, we will conduct a power analysis using the Cohen method [23, 24]. For its calculation we will use statsmodels.stats.power of the statsmodels Python package:

```
from statsmodels.stats.power \
import FTestAnovaPower
analysis = FTestAnovaPower()
k_groups = 3
n_per_group_large_75 = \
analysis.solve_power(effect_size=0.4, \
alpha=0.05, power=0.75, k_groups=3)
print(f"(f=0.4): {n_per_group_large_75:.1f}")
```

The calculated result is 57. When expecting a large effect, the sample is sufficient for the study because the power of subsets DS11, DS12, DS13 exceeds the obtained value.

4 RESULTS AND DISCUSSION

In this paper, we reviewed several API services that provide plagiarism checking and detection of text fragments built with the help of AI.

Turnitin is a well–known plagiarism detection service but is not directly accessible to individual users [25]. Access to Turnitin requires an institutional li-cense. If a university or educational institution does not hold such a license, its lecturers and students cannot use the platform for originality checking or academic integrity purposes. Thus, there is a need to consider other tools for detecting plagiarism and AI–generated text.

An important requirement for such services is the availability of an API that allows them to be integrated into the educational environment and a set of tools used by the teaching staff who evaluate students' work.

The result of our work is an API which combines existing services, its main processes are shown in Figures 1 and 2.

During the research process we conducted an AI detection using each of the services for data subsets DS11 (results are shown in Table 1), DS12 (Table 2) and DS13 (Table 3).

Table 1: Statistic results of AI detection services execution data subset DS11.

AI detection services	Average	Min	Max
GptZero	76	62	100
OpenAI	76.2	75	80
Writer	3.3	2.0	5/0
CopyLeaks	91.6	81.2	100
Sapling	0.1	0.0	0.2
Grammarly	0.0	0.0	0.0
ZeroGPT	51.4	39.9	62.8
Undetectable.ai	87.5	87.5	87.5
EdenAI	52.7	41.1	70.0

Table 2: Statistic results of AI detection services execution for data subset DS12.

AI detection services	Average	Min	Max
GptZero	76.6	58.0	93.0
OpenAI	77.0	65.0	85.0
Writer	1.2	0.0	3.0
CopyLeaks	80.0	0.0	100.0
Sapling	0.0	0.0	0.0
Grammarly	0.0	0.0	0.0
ZeroGPT	46.0	42.0	48.0
Undetectable.ai	83.8	68.8	85.0
EdenAI	53.8	38.7	68.1

Table 3: Statistic results of AI detection services execution for data subset DS13.

AI detection services	Average	Min	Max
GptZero	74.6	62.0	98.0
OpenAI	39.5	15.0	57.5
Writer	2.3	0.0	4.0
CopyLeaks	0.0	0.0	0.0
Sapling	0.0	0.0	0.0
Grammarly	0.0	0.0	0.0
ZeroGPT	40.5	29.0	48.5
Undetectable.ai	10.5	0.0	31.3
EdenAI	48.8	43.0	62.4

Our testing shows that Grammarly, Sapling and Writer do not allow detecting AI in text fragments. Possible reasons could be unsupported languages, specific subject areas of text fragments.

Results shown in Table 1 and 2 demonstrate that

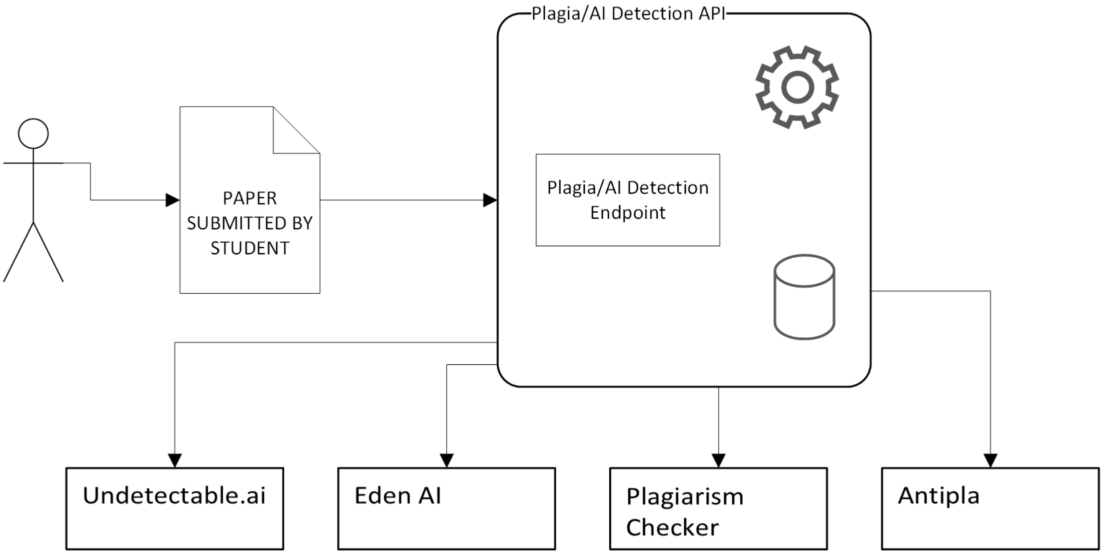

Figure 1: The process of initiating processing of a document uploaded by student.

differences between texts generated by GPT-4o and older versions of ChatGPT are not relevant.

GptZero, EdenAI results have a large number of false positives for papers which do not contain AI generated text. The best results are obtained using Undetectable.ai, there's only 10.5% false positives here. Results given by OpenAI and ZeroGPT are worse, but also acceptable.

The main conclusion is that there's no services which can be used as an automated tool for AI detection. The decision must be made by humans (lecturer which checks student work), AI detectors (or a combination of them) can only act as an advising mechanism.

In this paper we reviewed the following services for plagiarism detection: Eden AI [26], GPTZero [27], OpenAI [28], Grammarly [16, 29], ZeroGPT [30].

During the research processwe conducted a plagiarism detection using each of the services for data subsets DS21 (results are shown in Table 4), DS22 (Table 5) and DS23 (Table 6).

Results show that almost all the services do not generate false positives, there are issues with OpenAI only which tries to find sources even for original texts (Table 7). The same results were obtained for text fragments generated by AI – all services except-ing OpenAI found no plagiarism in them.

GptZero, Winston.ai and EdenAI results can be considered as good candidates for plagiarism detection because the percentages of errors are less than 11% (Table 7). It saves time for lecturers who check student papers.

In addition, these technologies are developing at a rapid pace and in a short time the service that is most optimal may change. At the same time, the task that needs to be solved by the lecturer who evaluates the student's work is quite stable. In most cases, information about the percentage of plagiarism and AI–generated content is sufficient.

That is why we decided to create our API and integrate it into services which are used for communication between students and lecturers in the educational process. The service we are developing is an aggregator and wrapper over a number of external services described above. Development stack is C#, ASP .NET.

An example of typical API that we use is EdenAI which provides a list of services, including Ai Detection and Plagia Detection [26].

AI Detection service is an aggregator of other services, so it supports Originality AI [31], WinstonAI [32], Sapling [33]. It is possible to specify which 3rd party services to use, it is important because the cost of operation depends on it. Response for each provider contains resulting ai_score and a list of text fragments with ai_score and prediction for each of them. A prediction is one of two values: "AI-generated" and "original". For each provider response may also contain original response from each provider.

Plagia Detection service supports 3rd party services such as Originality AI and Sapling. Response for each provider consists of the resulting plagia_score and a list of objects. Every object contains text and a list of candidates each consisting of url (where this fragment is placed), plagia_score, prediction ("plagia-

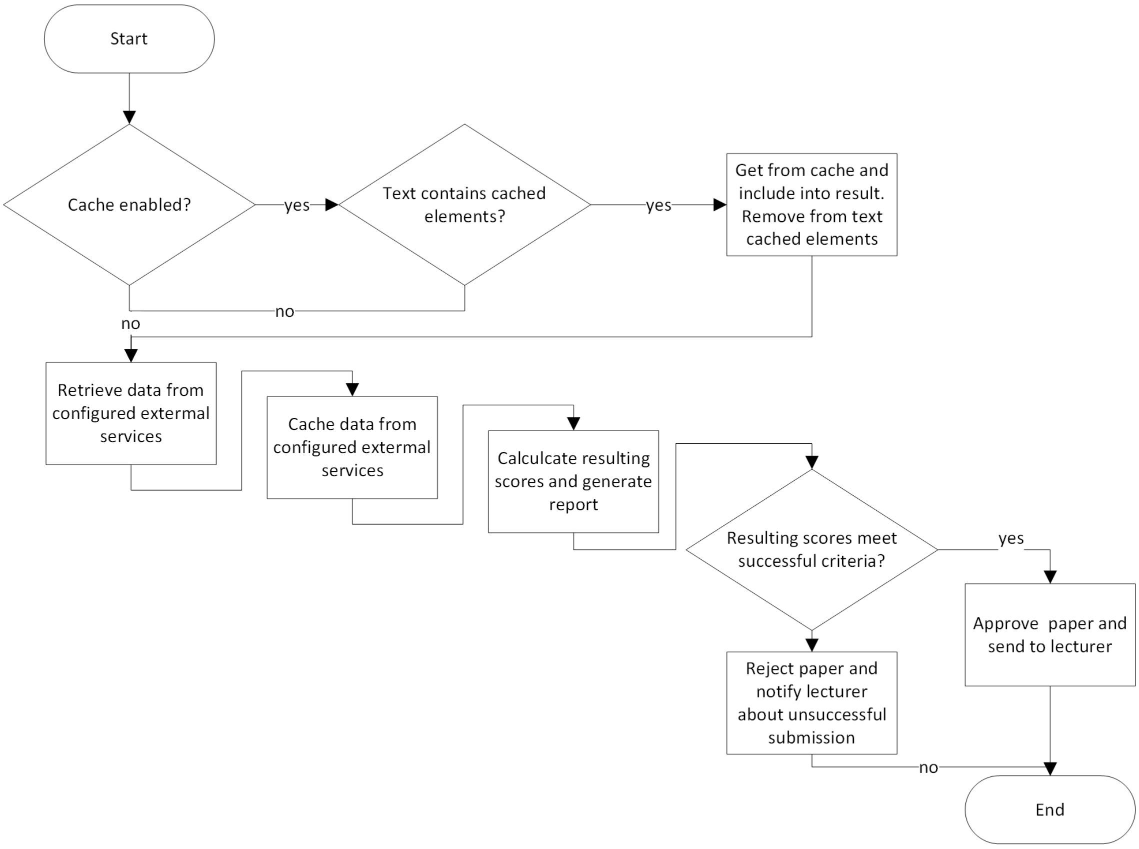
Start
Cache enabled?
yes
Text contains cached elements?
yes
Get from cache and include into result. Remove from text cached elements
no
no
Retrieve data from configured externmal services
Cache data from configured externmal services
Calculcate resulting scores and generate report
Resulting scores meet successful criteria?
yes
Approve paper and send to lecturer
Reject paper and notify lecturer about unsuccessful submission
no
End

Figure 2: The flowchart of AI/plagiarism detection in a document uploaded by a student.

Table 4: Statistic results of plagiarism detection services execution for data subset DS21.

Plagiarism detection services	Incorrect detections/errors	Incorrect detections/errors, %
Grammarly	113.0	84.3
GptZero	16.0	11.9
OpenAI	57.0	42.5
Plagiarism detector.net	53.0	39.6
Winston.ai	24.0	17.9
Originality.ai	127.0	94.8
EdenAI	24.0	17.9

Table 5: Statistic results of plagiarism detection services execution for data subset DS22.

Plagiarism detection services	Incorrect detections/errors	Incorrect detections/errors, %
Grammarly	0.0	0.0
GptZero	0.0	0.0
OpenAI	11.0	19.6
Plagiarism detector.net	0.0	0.0
Winston.ai	0.0	0.0
Originality.ai	0.0	0.0
EdenAI	0.0	0.0

Table 6: Statistic results of plagiarism detection services execution for data subset DS23.

Plagiarism detection services	Incorrect detections/errors	Incorrect detections/errors, %
Grammarly	0.0	0.0
GptZero	0.0	0.0
OpenAI	2.0	6.5
Plagiarism detector.net	0.0	0.0
Winston.ai	0.0	0.0
Originality.ai	0.0	0.0
EdenAI	0.0	0.0

Table 7: Statistic results of plagiarism detection services execution for all test cases.

Plagiarism detection services	Type 1 errors (false positives)	Type 2 errors (false negatives)	Type 1 errors, %	Type 2 errors, %	Errors, %
Grammarly	0.0	113.0	0.0	51.1	51.1
GptZero	0.0	16.0	0.0	7.2	7.2
OpenAI	13.0	57.0	5.9	25.8	31.7
Plagiarism detector.net	0.0	53.0	0.0	24.0	24.0
Winston.ai	0.0	24.0	0.0	10.9	10.9
Originality.ai	0.0	127.0	0.0	57.5	57.5
EdenAI	0.0	24.0	0.0	10.9	10.9

rized" or "original") and plagiarized_text.

Back to our API, the main endpoint is POST /v1/document. It allows the uploading of a document and starts a process of AI and plagiarism detection. The figure 1 shows the process of initiating document processing, it is triggered by a document submitted by student.

As it was mentioned above it should be possible to manage usage of external APIs – enable or disable them, set API key etc. For the current version of API we implemented this functionality based on application settings. Below is an example of appsettings.json fragment:

```
  "AIDetection": {
 "SuccesfulScore": 0.95,
 "Undetectable.ai": {
   "Enabled": true,
   "ApiKey": "***"
 },
 "EdenAI": {
   "Enabled": true,
   "ApiKey": "***",
   "Subservices": {
     "OriginalityAI": {
       "Enabled": true
     },
     "WinstonAI": {
       "Enabled": false
     },
     "Sapling": {
       "Enabled": true
     }
   }
 },
 ...
    },
  "PlagiaDetection": {
 "SuccesfulScore": 0.8,
 // settings are similar to AIDetection section
  },
  "cacheSettings": {
 "Enabled": true,
 "ExpirationMinutes": 1440
  }
```

It consists of 3 main sections: settings for AI detection, settings for plagiarism detection and cache settings. In an example above Undetectable.ai, and EdenAI (OriginalityAI and Sapling) APIs are used for AI detection. Antipla and EdenAI/WinstonAI are disabled in this configuration. SuccesfulScore = 0.95 means that 5% of paper is allowed to be detected as AI generated content.

Caching of responses allows to save locally results retrieved from external APIs. It allows to minimize costs needed for paper processing. Let's consider the situation when a student submits a paper, but it contains a high percentage of plagiarism. Some changes are applied to the next version of a paper, but the majority of it remains the same. Caching allows us to reuse results of processing from previous attempts. It means that external API will be called only for new or edited text fragments.

The flowchart of this process is shown on figure 2. This flowchart is simplified and just shows the concept. The real process is asynchronous, the initial endpoint finishes its work after asynchronous calls of external services. Responses from external services trigger the second part of flow.

5 CONCLUSIONS

In this paper, we reviewed the problem of plagiarism and AI detection in the educational process. Automation of this process allows lecturers to don't do routine work, but spend time on truly creative aspects.

We analyzed existing services for plagiarism and AI detection and tested them on a number of documents grouped by certain criterias. Based on our testing we consider GptZero, Winston.ai and EdenAI as good candidates for plagiarism detection. There are no services which demonstrate ideal results for AI detection, but from our point of view it can be achieved by recommendation based on a combination of Undetectable, OpenAI and ZeroGPT with a final decision made by a human (lecturer).

We developed an API that acts as an aggregator and wrapper for external APIs. A developed API can be integrated into tools and pipelines that are used for communication between students and lecturers. An API supports caching, it allows not to do additional calls retrieving the same data as on previous attempts. This API is configurable so it allows defining which services can be used. We provide a default config file based on the results of our testing, but it can be changed by a person who uses the API.

The proposed system helps reduce the time spent by the lecturer on checking student work and simplifies the process as a whole. This system is a convenient tool for maintaining academic integrity by checking the originality of student work.

Further research in this direction may focus on:

- multi-user support (currently software supports 1 API key per platform);
- integrate with existing educational platforms;
- add UI components for visualization of AI/plagiarism detection results.

REFERENCES

[1] D. Fanelli, "How many scientists fabricate and falsify research? A systematic review and meta-analysis of survey data," PLoS ONE, vol. 4, no. 5, 2009.

[2] J. Canzonetta, "Repurposing plagiarism detection services for responsible pedagogical application and (in)formative assessment of source attribution practices," Assessing Writing, vol. 50, p. 100563, 2021.

[3] F. R. Elali and L. N. Rachid, "AI-generated research paper fabrication and plagiarism in the scientific community," Patterns, vol. 4, no. 3, p. 100706, 2023.

[4] V. M. Fedorets, O. V. Klochko, I. A. Tverdokhlib, and O. A. Sharyhin, "Cognitive aspects of interaction in the 'Human - Artificial Intelligence' system," J. Phys.: Conf. Ser., vol. 2871, no. 1, p. 012023, Oct. 2024.

[5] N. Bhimrajka et al., "Privacy-preserving plagiarism checking," in Proc. 24th Int. Conf. Cryptol. India (INDOCRYPT 2023), Goa, India, 2023, pp. 105-125.

[6] N. Tahaei and D. C. Noelle, "Automated plagiarism detection for computer programming exercises based on patterns of resubmission," in Proc. 2018 ACM Conf. Int. Comput. Educ. Res., Espoo, Finland, 2018, pp. 178-186, [Online]. Available: https://doi.org/10.1145/3230977.3231006.

[7] S. Myronenko and Y. Myronenko, "Detecting plagiarism in student assignments using source code analysis," WSEAS Trans. Comput. Res., vol. 12, pp. 367-376, 2024.

[8] R. Maertens et al., "Discovering and exploring cases of educational source code plagiarism with dolos," SoftwareX, vol. 26, p. 101755, 2024.

[9] A. V. Slobodianiuk and S. O. Semerikov, "Advances in neural text generation: A systematic review (2022-2024)," CEUR Workshop Proc., vol. 3917, pp. 332-361, 2025.

[10] E. M. Hambi and F. Benabbou, "A new online plagiarism detection system based on deep learning," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 9, 2020, [Online]. Available: https://doi.org/10.14569/IJACSA.2020.0110956.

[11] K. Vani and D. Gupta, "Detection of idea plagiarism using syntax-semantic concept extractions with genetic algorithm," Expert Syst. Appl., vol. 73, pp. 11-26, 2017, [Online]. Available: https://doi.org/10.1016/j.eswa.2016.12.022.

[12] O. A. Sharyhin and O. V. Klochko, "Automation of checking student assignments in IT-related subjects based on AI systems," CEUR Workshop Proc., vol. 3781, pp. 88-97, 2024, [Online]. Available: https://ceur-ws.org/Vol-3781/paper07.pdf.

[13] V. I. Klochko, O. V. Klochko, and V. M. Fedorets, "Visual modeling technologies in the mathematical training of engineers in modern conditions of digital transformation of approaches to creating visual content," in Proc. 3rd Workshop Digit. Transform. Educ., Lviv, Ukraine, 2024, pp. 57-73, [Online]. Available: https://ceur-ws.org/Vol-3771/paper20.pdf.

[14] O. V. Klochko et al., "Anthropologically oriented strategies of interaction in the Human-Computer system," J. Phys.: Conf. Ser., vol. 2611, no. 1, p. 012018, Oct. 2023.

[15] Originality.ai, "7 Common Types Of Plagiarism - Examples And How To Avoid," 2025, [Online]. Available: https://originality.ai/blog/common-typesof-plagiarism.

[16] Grammarly Inc., "7 Common Types of Plagiarism, With Examples," 2024, [Online]. Available: https://www.grammarly.com/blog/plagiarism/typesof-plagiarism/.

[17] A. Benito, "Academic Integrity and the use of Artificial Intelligence (AI)," 2023, [Online]. Available: https://academicintegrity.org/resources/blog/119-2023/november-2023/471-academic-integrity-andthe-use-of-artificial-intelligence-ai.

[18] L. Volante, C. DeLuca, and D. A. Klinger, "ChatGPT and cheating: 5 ways to change how students are graded," World Econ. Forum, 2023, [Online]. Available: https://www.weforum.org/stories/2023/03/chatgptand-cheating-5-ways-to-change-how-students-aregraded/.

[19] ADG Creative, "International Center for Academic Integrity," 2023, [Online]. Available: https://academicintegrity.org/.

[20] S. Dann, M. Draper, and I. Glendinning, "Reflections on the European Network for Academic Integrity Conference," Qual. Assur. Agency, 2023, [Online]. Available: https://www.qaa.ac.uk/en/newsevents/blog/reflections-on-the-european-network-foracademic-integrity-conference.

[21] European Union, "AI Act," 2025, [Online]. Available: https://digitalstrategy.ec.europa.eu/en/policies/regulatoryframework-ai.

[22] Princeton University, "WordNet. A Lexical Database for English," 2025, [Online]. Available: https://wordnet.princeton.edu/.

[23] J. Cohen, Statistical Power Analysis for the Behavioral Sciences. Routledge, 2013.

[24] J. Buckley, "Conducting power analyses to determine sample sizes in quantitative research: A primer for technology education researchers using common statistical tests," J. Technol. Educ., vol. 35, no. 2, pp. 81-109, 2024.

[25] Turnitin LLC, "Higher education: Everything you need to uphold integrity at your institution," 2025, [Online]. Available: https://www.turnitin.com/solutions/higher-education.

[26] Eden AI, "The Full Stack AI Platform for Developers," 2025, [Online]. Available: https://docs.edenai.co/.

[27] GPTZero, "More than an AI detector," 2025, [Online]. Available: https://gptzero.me/.

[28] OpenAI, "Ask ChatGPT anything," 2025, [Online]. Available: https://openai.com/.

[29] Grammarly, "Grammarly: Responsible AI that ensures your writing and reputation shine," 2025, [Online]. Available: https://www.grammarly.com/.

[30] ZeroGPT, "AI Detector for ChatGPT, GPT-4o, Gemini, Claude AI Generated Text," 2025, [Online]. Available: https://zerogpt.org/.

[31] Originality.ai, "Originality AI Plagiarism and Fact Checker," 2025, [Online]. Available: https://originality.ai/.

[32] Winston AI, "The Most Trusted AI Detector," 2025, [Online]. Available: https://gowinston.ai/.

[33] Sapling, "AI Detector," 2025, [Online]. Available: https://sapling.ai/ai-content-detector.