Femenca Noronha, 2025, 13:1
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Detecting Text Similarity: A Machine Learning Approach to Plagiarism Checking

Femenca Noronha[1], Kaif Khan[2]
Department of Information Technology[1]
Department of Data Science, Thakur College of Science and Commerce
Thakur Village, Kandivali (East), Mumbai-400101, Maharashtra, India[1,2]

Abstract- This research introduces Plagiarism Checker, an advanced plagiarism detection system that utilises machine learning and NLP algorithms to efficiently detect textual similarities. Using TF-IDF for feature extraction and cosine similarity, the system ensures high accuracy in identifying plagiarism within documents. Correlating with this, Jaccard similarity and n-gram analysis can highlight common word patterns in documents as well as identify paraphrased text. Built with Flask, the web interface allows for the seamless upload of documents as well as analysis. To improve these accuracy findings, understanding the text pre-processing techniques such as tokenization, stopword removal, and lemmatization is important. The error of processing AI-generated text where obfuscation techniques are used is addressed within the research. Future updates will see the incorporation of deep learning models such as LSTM and transformers to enhance the detection capabilities of Plagiarism checkers. The contribution of the research to the advancement of automated plagiarism detection is to ensure originality as well as academic integrity.

Keywords- Natural Language Processing (NLP), Plagiarism Detection, Machine Learning, Text Similarity Detection, Deep Learning for Plagiarism Detection.

I. INTRODUCTION

Plagiarism detection has been of growing importance in ensuring academic integrity, originality, and ethical writing practices. With the fast growth of digital content and ease of access to numerous online resources, copying and re-structuring of text has become a widespread issue. Traditionally, methodologies used for detecting plagiarism are now inadequate as both intentional and unintentional forms of plagiarism, as well as copy and paste techniques and example texts and text delivery obfuscation, continue to become increasingly prevalent. Full-blown machine learning and NLP techniques, though, require developing automated plagiarism detection systems that utilize complex computational techniques to accurately identify copied or restructured text provided over networks. This research introduces Plagiarism Checker, an NP-complex machine learning and NLP-powered plagiarism detection system that is designed to analyze large volumes of text efficiently.

The system uses TF-IDF Term Frequency – Inverse Document Frequency For feature extraction and cosine similarity to compare document similarities as Jaccard similarity and n-gram analysis and common word patterns. To enhance detection accuracy, different text pre-processing guidelines, such as tokenization, stopword removal, and lemmatization, are utilized within the system. Flask is used to develop a user-friendly web interface that enables standard text documents to be uploaded and real-time analysis. Beyond classical text

matching, this study also addresses the issues in detecting AI-generated content and techniques used to obfuscate or disguise copied text.

Future enhancements are aimed at the inclusion of deep learning models, including LSTM (Long Short-Term Memory) and transformers, to ensure a better detection accuracy of cases of plagiarism that span a complex set of sub-ordinates questions. Combining traditional and modern computational approaches is envisaged to greatly contribute to the discipline of automated detection of plagiarism by incorporating review and analysis of documents about academic integrity and original content in digital mediums.

II. LITERATURE REVIEW

Direct text matching, cryptographic hash functions, and structural comparisons are used by Meyer zu Eissen et al.(1) in traditional methods to identify similarities. However, these approaches struggle against obfuscation and AI-generated content. A variety of advances, for example, machine learning and semantic textual analysis, have led to enhancements in detection accuracy by detecting writing style differences and linguistic attributes. Detection of intrinsic plagiarism using a single document without considering external references has been in focus over the last years due to its capability to flag inconsistencies in the writing style that could be applied to scenarios where source information is not available (Meyer zu Eissen Stein, 2006). Future studies should be dedicated to hybrid methods (adding AI-based models, semantic similarity calculations, and dynamic code assessment) to improve plagiarism detection in coding tasks.

Donaldson et al.(2) investigated the design and implementation of an automated method of detecting structural similarities in student programs that are used to combat common techniques used by students to dilute the originality of code by renaming variables and changing formatting styles. However, traditional countermeasures to AI-generated content have thus far had limited success due to the emergence of AI-assisted content plagiarism (Wyk et al., 2020) as underlying the method to bypass existing detection schemes (Khalil and Er, 2023). Advanced computational methods such as machine learning, syntax analysis, abstract syntax trees, and learning algorithms have increased accuracy in this area. However, as obfuscation strategies employed by AI continue to evolve, methods to compromise existing detection systems necessitate hybrid approaches that integrate AI-driven solutions with static and dynamic analysis in static and dynamic execution tracing to increase reliability in programming education.

Potthast et al. (3) propose an evaluation framework for plagiarism detection using the PAN-PC-10 corpus, which includes real, simulated, and artificial plagiarism cases. The work highlights the weaknesses of existing evaluation schemes and stresses the need for a standardized benchmark to conduct plagiarism detection. The study will classify detection methods into intrinsic and external approaches, which will address problems associated with computational methods as well as text obfuscation challenges. Key metrics like precision, recall, and detection granularity will assess the effectiveness of the algorithm, which will be used to improve detection improvements and recommend linguistic and text analysis techniques that are advanced to improve accuracy as well as reliability as mentioned in large-scale corpora as well as automation.

Khaled et al. (4) provide an overview of various plagiarism detection methods and tools with emphasis on the challenges that are posed by verbatim and intelligent plagiarism. Plagiarism is categorized into monolingual and cross-lingual forms, and some of the established detection techniques include character-based, syntax-based, and semantic-based techniques. In addition to this, the researchers discuss the evolution of plagiarism detection software such as Turnitin and PlagScan and stress the increasing need for automation of detection due to the inability to properly manage the Specter of manual methods. The paper examines the effectiveness of datasets such as WordNet and PAN when training detection

systems. As the detection of plagiarism evolves in the future, particularly valuable research should focus on integrating concepts of AI and Deep Learning for more accurate and efficient detection of plagiarised content.

Naik et al. (5) deliver a very detailed analysis of plagiarism detection tools addressing issues such as copy-paste specimens, disguised detection, and translation-based production of plagiarism. Detection methods are divided into external, where facing documents are compared to databases, and internal, where the writing style of a particular paper is analyzed to find inconsistencies. Commonly employed tools Turnitin, PlagScan, and Grammarly detect text-based plagiarism, while MOSS and JPLAG provide code similarity analysis only. The paper also highlights emerging approaches for the detection of paraphrased and structured modified content in AI technologies, such as machine learning and natural language processing, which will be used to identify content produced by paraphrase or transformation of structure. Despite advancements in this technology, detection accuracy will vary, necessitating the development of continuous improvements in the AI algorithms used. Future work should reside on integration of deep learning and NLP processing to enhance the precision and reliability of plagiarism detection of systems.

Foltýnek et al.(6) conducted a systematic review of 239 research papers published between 2013 and 2018, examining major developments in the detection of plagiarism. The study categorizes methods for detection into intrinsic and extrinsic. It is stoic that improvements to semantic text analysis performance, as well as non-textual content evaluation, have been made alongside improvements in machine learning applications. The paper also discusses problems such as the inability to automatically detect plagiarism that is highly obfuscated and the lack of thorough methodological evaluations of the quality of existing detection systems. Even though traditional text-matching pursuit tools are widely used in practice, researchers suggest the need for hybrid detection models so that they include linguistic and syntactic as well as semantic analysis. Future research should concentrate on integrating heterogeneous detection techniques with AI and deep learning to improve the performance and reliability of plagiarism detection systems.

III. METHODOLOGY

The Plagiarism Checker system makes efficient use of a combination of natural language processing (NLP) techniques and machine learning algorithms to ferret out plagiarism from documents using a methodology that starts with data gathering and input processing, where the system will allow texts to be uploaded in multiple documents formats such as.txt,.pdf and.docx after which they will be converted into plain text for further analysis. The next step in the methodology is the text feature extraction; this involves tokenization, which involves breaking text into words or phrases, removal of stopwords (words that do not add meaning and thus take up space in text documents that do not contribute anything), lemmatization (reducing words to their root form), and case conversion to ensure that the text is converted into a uniform state for comparison. Once the content has been pre-processed, feature extraction procedures of TF-IDF and n-gram analysis are applied, against which the system can form numerical representations for the textual data. These steps capture word importance and contextual similarity between documents and ensure that plagiarism is detected. The system is designed to ascertain the level of document similarity by employing a variety of similarity algorithms, which include the cosine similarity algorithm that evaluates the angle between a document vector and the vector that represents all other documents. Additionally, Jaccard similarity is employed whereby all words that are utilized in multiple documents are identified and provided as a comparison of the documents whereby the output of this can be regarded as the degree of similarity between documents based upon the number of terms that are shared between two document arrays. The final element of the system is the use of Euclidean distance, whereby the differences in the numerical order between document vectors are measured,

thus producing an estimate of the similarity between documents.

By using these solutions, the system can detect directly copied content, paraphrased text, and structurally modified plagiarism. The Flask web framework is used for deploying the web-based system, which allows the user to upload documents and perform plagiarism checks alongside visual similarity reports. The interface provides a user-friendly platform for document analysis; the platform allows the user to see flagged plagiarized sections along with the percentage of similarity generated. Although the Plagiarism checker is currently very effective in detecting AI-generated content and text obfuscation, advances are required to progress fully. The upcoming improvements will instead focus on using deep learning tools such as LSTM networks and transformer models to improve the accuracy of detection and adaptability of machine detection by increasing the robustness of detection, ensuring the effectiveness of the system. Also, refinement of text analysis techniques will provide the ability to differentiate between instances where words are changed but the meaning remains the same by using semantic analysis. By overcoming these roadblocks, the system will be able to improve its robustness as well as the intelligence of detection tools, ensuring academic integrity and preserving originality in digital content.

Fig 1:- Plagiarism Detection Process

IV. RESULT & DISCUSSION

The Plagiarism checker model was evaluated on a dataset that had 370 entries containing both 187 non-plagiarized text entries and 183 plagiarized entries. Preprocessing steps were applied to all samples, including the removal of punctuation, making words lowercase, and elimination of stop words to ensure consistency in the text representation. Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was used for feature extraction to allow the model to capture the level of importance of words in the detection of plagiarism. Several machine learning models were trained and tested with a Support Vector Machine (SVM), achieving the highest accuracy of 87. 84% outperforming Multinomial Naïve Bayes (accuracy of 86. 49%, Logistic Regression (accuracy of 82. 43% and Random Forest (accuracy of 79. 73%.

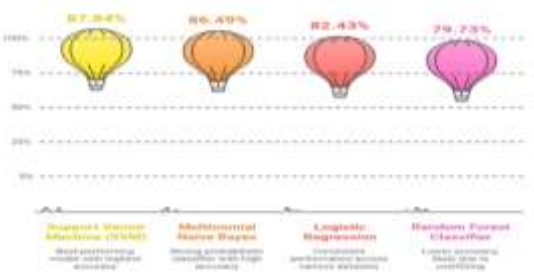

Fig 2:- Comparison of ML model accuracy

Data exploratory analysis led to invaluable learning about the text values through the use of visualization aids. The distribution of text values indicates a balanced dataset this allows the fairness of the model to be maintained when training the data.

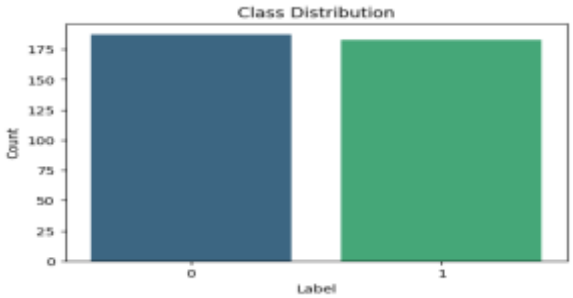

Fig 3:- Class Distribution

The word cloud highlights frequently occurring words in the text, giving information as to the common patterns found within poorly written and those samples that are non-plagiarized text.

Fig 4:- Word Cloud of Source Text

Analysis of the text length distribution found that most of the documents contain between 200 - 300 characters, the result of which prompted text preprocessing strategies.

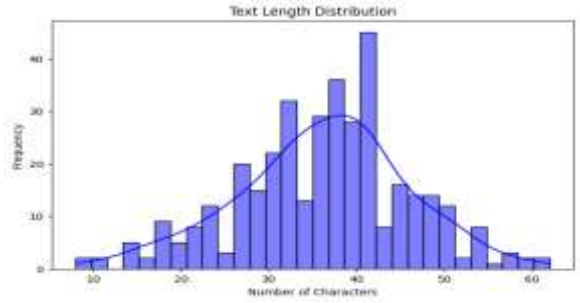

Fig 5:- Text Length Distribution

In addition, it was also illustrated by a graph of the TF-IDF feature importance as to the top 20 words that contributed the most to the effectiveness of the feature vector obtained.

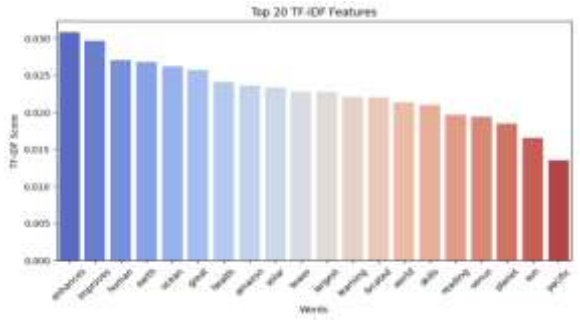

Fig 6:- Word Cloud of Source Text

The SVM model and trained TF-IDF vectorizer were stored using pickle for real-world deployment of the system. For the study, it has been shown that the SVM has good results for plagiarism classification, which means it is suitable for academic and professional plagiarism detection. However, it must be pointed out that the current system is not able to detect all types of text, including that generated by AI and Paraphrased text, and due to this, future research is needed to develop techniques like LSTM and transformers however, it is critical in the detection accuracy of the system. The ability of the system to automatically save and reload models enhances the usability of the method, and it is a promising solution in automated plagiarism detection.

V. CONCLUSION AND FUTURE SCOPE

The Plagiarism Checker system showed the benefit of machine learning and natural language processing (NLP) techniques for detecting plagiarism effectively. Through TF - IDF feature extraction and similarity measures such as cosine similarity, Jaccard similarity, and n-gram analysis, the system was able to accurately detect direct copying, paraphrased content, and text obfuscation techniques. The SVM model was found to have a much higher accuracy score when applied to other machine learning classifiers due in part to the high accuracy score of 87.84 achieved by the SVM model, making it the most suitable for the detection of plagiarism.

This gave useful visual impressions when visual analysis was conducted on terms such as the class distribution, word cloud, text length distribution, and TF - IDF feature importance, which made use of the n-gram model to give beneficial insights into the dataset in terms of its structure and the influence of the features of the text. The Flask app-based Flask-restful interface integrated was able to process document submissions smoothly and allow for real-time plagiarism analysis to be completed, making it an ideal solution for both the academic and professional fields. However, further advancements were required to address issues such as discriminating against AI-generated content and heavily paraphrased text, and there was a need for improvement.

Enhancements in the future will have to do with the incorporation of an LSTM (Long Short-Term Memory) deep learning model and transformers (BERT, GPT) to facilitate increased detection accuracy for paraphrases and AI-generated text.

Also, the application of semantic analysis methods can significantly improve the ability of a system to comprehend contextual similarities that go beyond direct word comparison. Adding to the diversity of the dataset by including academic texts, programming blocks, and AI scripts will enhance model performance even further.

Creating a real-time API to support plagiarism detection can allow effortless embedding in LMS, other educational institutions, and even publishing houses. Ultimately, advanced methods of explanation and understanding in terms of the results produced by the plagiarism detection system would allow the users to comprehend the rationality of flagging a document, thus increasing transparency and effectiveness. Such developments will further allow the building of a more extensible and intelligent plagiarism detection system that upholds the academic honesty and originality of a written work.

REFERENCES

1. Eissen SM, Stein B. Intrinsic plagiarism detection. In Advances in Information Retrieval: 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, 2006. Proceedings 28 2006 (pp. 565-569). Springer Berlin Heidelberg.
2. Donaldson JL, Lancaster AM, Sposato PH. A plagiarism detection system. InProceedings of the twelfth SIGCSE technical symposium on Computer science education 1981 Feb 1 (pp. 21-25).
3. Potthast M, Stein B, Barrón-Cedeño A, Rosso P. An evaluation framework for plagiarism detection. InColing 2010: Posters 2010 Aug (pp. 997-1005).
4. Khaled F, Al-Tamimi MS. Plagiarism detection methods and tools: An overview. Iraqi Journal of Science. 2021 Aug 31:2771-83.
5. Naik RR, Landge MB, Mahender CN. A review on plagiarism detection tools. International Journal of Computer Applications. 2015 Sep;125(11):16-22.
6. Foltýnek T, Meuschke N, Gipp B. Academic plagiarism detection: a systematic literature review. ACM Computing Surveys (CSUR). 2019 Oct 16;52(6):1-42.
7. Lukashenko R, Graudina V, Grundspenkis J. Computer-based plagiarism detection methods and tools: an overview. InProceedings of the 2007 international conference on Computer systems and technologies 2007 Jun 14 (pp. 1-6).
8. Khalil M, Er E. Will ChatGPT G et You Caught? Rethinking of Plagiarism Detection. InInternational Conference on Human-Computer Interaction 2023 Jun 9 (pp. 475-487). Cham: Springer Nature Switzerland.
9. Zimba O, Gasparyan A. Plagiarism detection and prevention: a primer for researchers. Reumatologia/Rheumatology. 2021 May 13;59(3):132-7.
10. Kustanto C, Liem I. Automatic source code plagiarism detection. In2009 10th ACIS International conference on software engineering, artificial intelligences, networking and parallel/distributed computing 2009 May 27 (pp. 481-486). IEEE.