

Plagiarism Detection Using Machine Learning Techniques

Sudhakar, Adinath, Ashfaq, Inayat¹ Dr. Basavalinga Swamy² Dr. Vinita Patil³

¹UG Students department of Electronics & Communication

²Faculty, department of Electronics & Communication

³Professor, Department of ECE, Lingaraj Appa Engineering College Bidar
Lingaraj appa engineering college bidar, Karnataka India.

Emails: basavalingaswamy2020@gmail.com

ABSTRACT

Plagiarism has become a major concern in academic and professional environments due to the easy availability of digital content. Traditional plagiarism detection methods such as keyword matching fail to identify paraphrased or semantically similar text. This paper proposes an intelligent plagiarism detection system using machine learning and Natural Language Processing techniques. The system preprocesses input documents, retrieves relevant online sources using Google Custom Search API, and measures similarity using TF-IDF vectorization and cosine similarity. The proposed approach effectively detects both direct and rephrased plagiarism. Experimental results demonstrate improved accuracy and reliability compared to conventional methods, making the system suitable for educational and content verification applications.

KEYWORDS: Plagiarism Detection, Machine Learning, TF-IDF, Cosine Similarity, Natural Language Processing, Google Search API

1. INTRODUCTION

Plagiarism refers to the unauthorized use of another person's work without proper acknowledgment. With the rapid growth of online resources, plagiarism has increased significantly. Existing tools often fail to detect paraphrased content or synonym-based duplication. Machine Learning (ML) provides advanced techniques for identifying hidden similarities in text. By combining NLP preprocessing, feature extraction, and similarity computation, plagiarism traditionalrule-based approaches.

This work presents a plagiarism detection.system that compares user-submitted documents against online sources and produces a similarity score with detailed reporting.

II. OBJECTIVE:

To perform effective text preprocessing and feature extraction Before analysing the text, it is essential to clean and process it. This includes steps like tokenization, stop-word removal, and stemming or lemmatization. The system will use feature extraction methods such as TF-IDF or word embeddings to convert the text into a numerical format suitable for machine learning algorithms.

To apply similarity metrics and classification models. The project will use similarity algorithms like Cosine Similarity or Jaccard Similarity to compare texts.In addition, machine learning models such as Support Vector Machine (SVM) or Logistic Regression may be trained to classify whether text pairs are plagiarized or original.

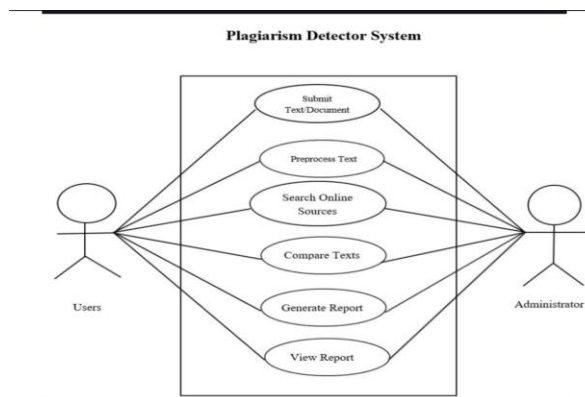
III. WORK ANALYSIS

Several approaches have been proposed for plagiarism detection:

- Sonawane and Prabhudeva used Karp-Rabin and string-matching algorithms achieving over 85% precision.
- Chitra and Rajkumar applied paraphrase recognition for sentence-level plagiarism detection.
- Engels et al. introduced deep learning with Doc2Vec embeddings for plagiarism classification.
- Awale et al. employed XGBoost for programming assignment plagiarism with 92% accuracy.

These studies highlight the importance of semantic similarity detection beyond surface-level matching.

IV. BLOCK DIAGRAM



V. METHODOLOGY

1. Problem Definition

Define the problem of plagiarism in digital content (academic, online, etc.). Set the goal: to develop an automated system that detects similarity between documents and identifies plagiarized content.

2. Data Collection

Collect a dataset of text documents, pairs of sentences, or student assignments. If no labeled dataset is available, synthetic data can be created by rephrasing existing content to simulate plagiarism.

3. Data Preprocessing

Preprocessing is essential to convert raw text into usable input for the machine learning model. Steps include: Tokenization: Splitting text into words or tokens. Lowercasing: Converting all text to lowercase to maintain consistency. Stop-word Removal: Removing common words (like "is", "the", etc.) that do not affect meaning.

Stemming/Lemmatization: Reducing words to their base/root form. Punctuation and Special Character Removal.

4. Feature Extraction

Convert processed text into numerical vectors that machine learning models can understand. Use TF-IDF (Term Frequency-Inverse Document Frequency) to transform documents into feature vectors. Alternatively, Bag of Words or word embeddings (Word2Vec, GloVe) can also be used.

5. Similarity Measurement

Calculate the cosine similarity between TF-IDF vectors to measure the degree of similarity between documents. A threshold (e.g., 0.8) is set to determine if documents are plagiarized.

6. Model Training (Optional for Classification-based Approach)

If you treat it as a supervised learning problem: Use algorithms like Logistic Regression, SVM, or Random Forest. Split dataset into training and testing sets. Train the model on labeled examples of plagiarized and nonplagiarized pairs.

7. Testing and Evaluation

Evaluate model accuracy using metrics such as: Precision Recall F1 Score Confusion Matrix Fine tune the model or threshold to improve accuracy.

8. Deployment (Optional)

Implement a simple GUI using Tkinter or a web app using Flask/Django. Users can upload documents and check for plagiarism.

VI. Results and Discussion

The system generates:

Plagiarism percentage score

Highlighted plagiarized content

Matched online sources. Graphical visualization(pie/bar chart) Results show that cosine similarity with TF-IDF performs well for both exact and partially paraphrased plagiarism.

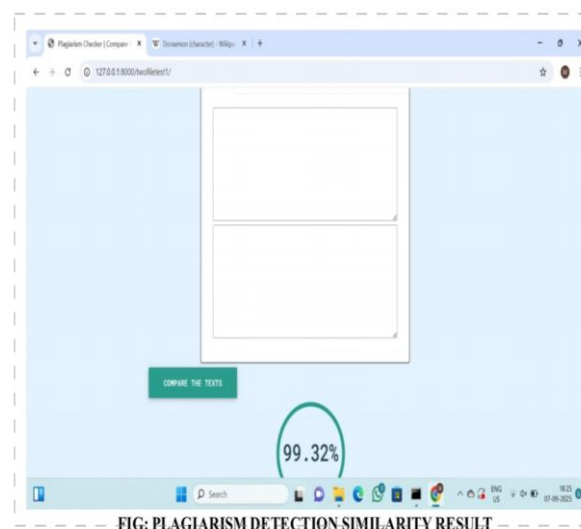


FIG: PLAGIARISM DETECTION SIMILARITY RESULT

The plagiarism detector developed using machine learning and Google API successfully identifies similarities between input documents and online resources. The cosine similarity algorithm ensures accurate text comparisons. This project provides a scalable and cost-effective alternative to commercial plagiarism detection tools. In this project, we created a plagiarism detection system using machine learning that can effectively identify copied or similar content in documents. By analyzing text with natural language processing techniques, the system is able to catch both exact matches and paraphrased content, making it more reliable than simple keyword-based tools. The project shows that machine learning can make plagiarism detection faster and more accurate, helping students, writers, and professionals maintain originality in their work. With future improvements like deeper semantic understanding and support for multiple languages, such a system can become even smarter and more widely useful. Overall, this project emphasizes the importance of technology in promoting ethical writing and preserving intellectual honesty

REFERENCES

- [1] K. Sonawane and P. S. Prabhudeva, "Plagiarism Detection Using Karp-Rabin Algorithm," International Journal of Computer Science, 2015.
- [2] A. Chitra and A. Rajkumar, "Plagiarism Detection Using Paraphrase Recognition," IEEE Conference, 2014.
- [3] S. Engels, V. Lakshmanan, and M. Craig, "Feature-Based Neural Networks for Plagiarism Detection," IEEE Transactions, 2018.