AI-Powered Plagiarism and AI Text Detection System for Regional Languages

Ms. Asma Mulla[1], Prasad Nagnath Kshirsagar[2], Gaurav Dipak Sutar[3], Vishal Ravindra Jarande[4], Sanket Nikesh Anpat[5], Nandini Umakant Shrimangle[6]
[12346]*Department of Computer Science and Engineering*
[123456] *Yashoda Technical Campus, Satara Faculty Of Engineering*

Abstract- In today's digital world, the use of online content and artificial intelligence tools has increased rapidly. Many students and writers use AI tools to generate text or copy content from different sources. This creates a serious problem in maintaining originality and academic honesty. Most existing plagiarism detection tools mainly support English and are not effective for regional languages like Marathi and Hindi. Also, they are not able to properly detect AI-generated text. This project presents an AI-powered system that can detect both plagiarism and AI-generated text in multiple languages, including Marathi, Hindi, and English. The system uses Natural Language Processing (NLP) and Machine Learning techniques to analyze the input text and provide accurate results. For plagiarism detection, the system uses methods like TF-IDF, n-grams, and semantic embeddings generated by advanced models such as IndicBERT and LaBSE. These techniques help in identifying not only exact copied text but also paraphrased and translated content across different languages. For AI text detection, the system uses perplexity and stylometric features such as sentence length, word usage, and punctuation patterns. These features help in identifying whether the text is written by a human or generated by AI tools. The system is designed with a user-friendly interface where users can enter or upload text. After processing, it generates a detailed report showing plagiarism percentage, AI probability score, and matched content if available. This project helps educational institutions, researchers, and content creators to verify the originality of text and maintain academic integrity. In the future, the system can be improved by adding more regional languages, increasing dataset size, and enhancing real-time detection capabilities.

Index Terms- Plagiarism Detection, AI-generated Text Detection, Natural Language Processing, Machine Learning, Multilingual System, IndicBERT, LaBSE, TF-IDF, Stylometric Analysis, Perplexity, Regional Languages.

I. INTRODUCTION

In recent years, the use of digital content has increased very rapidly in education, research, and online platforms. Students, researchers, and content writers often depend on the internet to collect information for assignments, reports, and articles. However, this easy availability of content has also increased the problem of plagiarism, where users copy text from different sources without giving proper credit. Plagiarism affects the originality of work and reduces academic integrity. Therefore, detecting copied content has become very important in educational institutions and professional fields [1].

At the same time, the development of Artificial Intelligence (AI) tools such as ChatGPT and other text generation systems has made it very easy to generate human-like text within seconds. While these tools are helpful, they also create challenges in identifying whether a piece of content is written by a human or generated by AI. Traditional plagiarism detection tools are not designed to detect AI-generated text, which creates a new problem in maintaining fairness and authenticity [2].

Most existing plagiarism detection systems mainly support the English language and are not effective for regional languages like Marathi and Hindi. India is a multilingual country, and a large amount of content is created in regional languages. These systems fail to detect translated or paraphrased content across different languages, which makes plagiarism detection more difficult. Therefore, there is a strong need for a system that can work across multiple languages and detect both direct and indirect plagiarism [3].

To solve these problems, this project proposes a multilingual AI-based system that can detect both plagiarism and AI-generated text. The system uses Natural Language Processing (NLP) and Machine Learning techniques to analyze text and understand its meaning. Techniques such as TF-IDF, n-grams, and transformer-based models like IndicBERT and LaBSE are used to identify similarity between texts. These methods help in detecting not only exact matches but also paraphrased and cross-language plagiarism [4].

In addition to plagiarism detection, the system also focuses on identifying AI-generated text using features like perplexity and stylometric analysis. These techniques analyze writing patterns such as sentence structure, word usage, and punctuation to determine whether the text is human-written or machine-generated. By combining both plagiarism detection and AI detection, the system provides a complete solution for ensuring originality and maintaining academic honesty [5].

II. LITERATURE REVIEW

In Plagiarism detection has been an important research area for many years. Early systems mainly focused on detecting exact copying of text using simple string matching and keyword comparison techniques. These methods were easy to implement but were not effective in detecting

paraphrased or translated content. Researchers later introduced more advanced approaches such as fingerprinting and vector space models to improve accuracy. However, these methods still had limitations when dealing with semantic similarity and cross-language plagiarism [6].

With the advancement of Natural Language Processing (NLP), more powerful techniques were developed for text analysis. TF-IDF (Term Frequency–Inverse Document Frequency) became one of the most commonly used methods for measuring the importance of words and finding similarity between documents. N-gram models were also introduced to capture sequences of words and detect partial matches in text. These approaches improved plagiarism detection but still struggled with understanding the actual meaning of the text [7].

In recent years, deep learning and transformer-based models have significantly improved text understanding. Models like BERT and its variants can capture the semantic meaning of sentences and provide better similarity detection. For multilingual tasks, models such as IndicBERT and LaBSE were developed to support Indian and multiple global languages. These models allow cross-language comparison, making it possible to detect plagiarism even when the text is translated from one language to another [8].

Apart from plagiarism detection, researchers have also worked on detecting AI-generated text. One important method is perplexity, which measures how predictable a piece of text is. AI-generated text usually follows a more predictable pattern compared to human writing. Another approach is stylometric analysis, which studies writing style features such as sentence length, vocabulary usage, and punctuation patterns. These features help in distinguishing between human-written and machine-generated content [9].

Recent studies have focused on combining multiple techniques to improve accuracy. Hybrid systems that use both lexical methods (like TF-IDF) and semantic methods (like embeddings) have shown better performance. Similarly, combining stylometric features with machine learning models has improved AI text detection. These advancements suggest that an integrated approach is more effective for handling complex problems like multilingual plagiarism and AI-generated text detection [10].

III. KEY FINDINGS

The study and implementation of this project have led to several important findings related to plagiarism detection and AI-generated text detection in multilingual environments. The key findings are explained below:

A. Need for Multilingual Plagiarism Detection

It is observed that most existing plagiarism detection tools mainly focus on the English language and do not perform well for regional languages like Marathi and Hindi. This creates a major gap in academic systems where students submit work in different languages. A multilingual system is necessary to ensure fairness and equal evaluation for all users [11].

B. Limitation of Traditional Methods

Traditional plagiarism detection techniques such as keyword matching and string comparison are not sufficient for modern requirements. These methods fail to detect paraphrased or reworded content. They also cannot identify plagiarism when the text is translated into another language. This shows the need for more advanced semantic-based techniques [12].

C. Importance of Semantic Understanding

The use of embedding models like IndicBERT and LaBSE helps in understanding the actual meaning of the text rather than just matching words. These models convert sentences into numerical vectors, allowing comparison based on meaning. This approach improves the detection of paraphrased and cross-language plagiarism significantly [13].

D. Effectiveness of Hybrid Approach

Combining multiple techniques such as TF-IDF, n-grams, and embeddings provides better results compared to using a single method. The hybrid approach increases accuracy by capturing both lexical similarity (word-level) and semantic similarity (meaning-level). This combination makes the system more reliable [14].

E. AI-generated Text Detection is Necessary

With the rapid growth of AI tools like ChatGPT, detecting AI-generated content has become very important. It is found that traditional plagiarism tools cannot identify AI-written text because it is original but machine-generated. Therefore, a separate AI detection module is required [15].

F. Role of Perplexity in AI Detection

Perplexity is an important measure used to detect AI-generated text. AI-generated content is usually more predictable and structured compared to human writing. By analyzing perplexity scores, the system can identify whether the text is likely generated by a machine.

G. Stylometric Features Improve Detection

Stylometric analysis, which studies writing style features such as sentence length, punctuation usage, and vocabulary patterns, plays a key role in AI detection. Human writing is usually more diverse, while AI-generated text often follows consistent patterns. This difference helps in classification.

H. Challenges Due to Limited Dataset

One major finding is that there is a lack of high-quality datasets for regional languages like Marathi and Hindi.

This affects the training and performance of machine learning models. More data collection is required to improve system accuracy.

I. Computational Cost of Advanced Models

Transformer-based models like IndicBERT and LaBSE provide better results but require high computational resources such as GPU. This can increase processing time and system cost, especially for large-scale applications.

J. Overall System Effectiveness

The combined system of plagiarism detection and AI text detection provides a complete solution for maintaining originality. It helps in improving academic integrity, detecting misuse of AI tools, and supporting multilingual content analysis effectively.

IV. SYSTEM ARCHITECTURE

The system architecture defines how different components of the project are organized and how they work together to perform plagiarism and AI-text detection. The system is designed in a modular way so that each part performs a specific function efficiently.

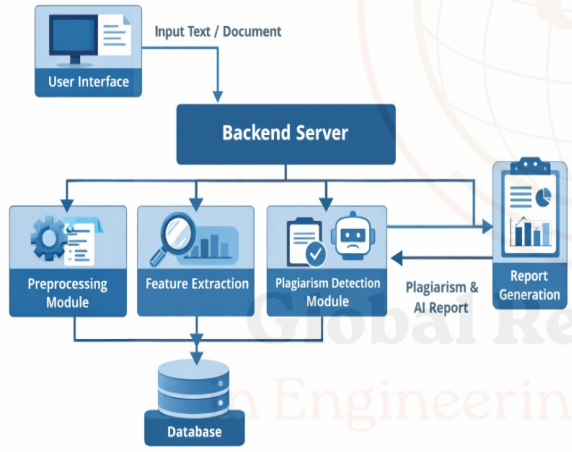


Fig.1-Systyem Architecture Diagram

The major components of the system are explained below:

A. User Interface (Frontend Layer)

The user interface is the first point of interaction between the user and the system. It is designed to be simple and user-friendly. The user can enter text or upload a document for checking. The frontend sends the input data to the backend server for processing and later displays the results in a clear format such as percentages and highlighted text [16].

B. Backend Server (Application Layer)

The backend server handles all the processing requests. It receives the input text from the frontend and manages the flow of data between different modules. It is responsible for calling various functions such as preprocessing, feature extraction, and detection modules. Technologies like Flask or FastAPI can be used to build the backend system [17].

C. Input Processing Module

This module is responsible for preparing the input text for analysis. It performs several preprocessing steps such as:

a. Removing special characters
b. Converting text to lowercase
c. Tokenization (splitting text into words)
d. Removing stopwords

This step ensures that the text is clean and ready for further processing, which improves the accuracy of the system [18].

D. Feature Extraction Module

After preprocessing, important features are extracted from the text. These features are used by the detection models. The system uses:

a. TF-IDF for word importance
b. N-grams for phrase matching
c. Embeddings using IndicBERT and LaBSE for semantic understanding

These features help the system to analyze both the structure and meaning of the text [19].

E. Plagiarism Detection Module

This module checks whether the input text is copied or similar to existing content. It compares the extracted features with stored or online data using similarity measures such as cosine similarity. It can detect:

a. Exact copying
b. Paraphrased text
c. Cross-language similarity

The output of this module is the plagiarism percentage along with matched content if available.

F. AI Text Detection Module

This module determines whether the text is written by a human or generated by AI. It uses:

a. Perplexity to measure predictability of text
b. Stylometric features such as sentence length and punctuation

A machine learning model is used to classify the text. This module works independently but its result is combined with plagiarism detection [20].

G. Database (Optional Storage Layer)

The database is used to store:

a. Text datasets
b. Preprocessed data

c. Previous results

It helps in faster comparison and improves system performance. Databases like MongoDB or PostgreSQL can be used.

H. Report Generation Module

After both detection processes are completed, this module generates a final report. The report includes:

a. Plagiarism percentage
b. AI probability score
c. Highlighted similar text

The results are presented in a simple and understandable format for users.

I. Workflow of the System

The complete system works in the following steps:

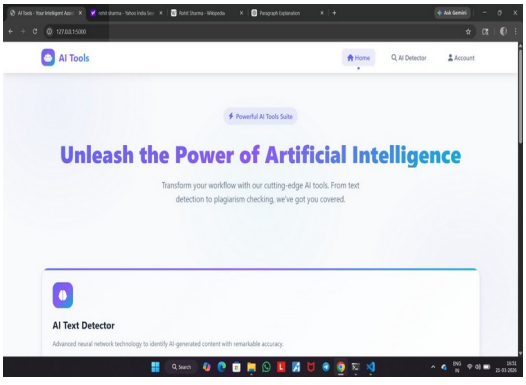
1. User inputs text through the interface
2. Text is sent to backend server
3. Preprocessing is performed
4. Features are extracted
5. Plagiarism detection is executed
6. AI detection is executed
7. Results are combined
8. Final report is displayed

J. Overall Architecture Design

The system follows a modular and scalable design. Each module works independently but is connected through the backend. This design makes the system easy to maintain, upgrade, and extend in the future. It also allows integration of additional languages and advanced models.

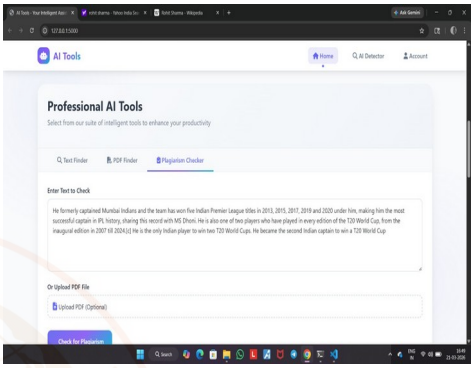
V. RESULTS AND DISCUSSION

The results of this project show that the proposed system is effective in detecting both plagiarism and AI-generated text in multiple languages such as Marathi, Hindi, and English. The system was tested using different types of input text including original content, copied text, paraphrased text, and AI-generated text. The performance of the system is discussed below.



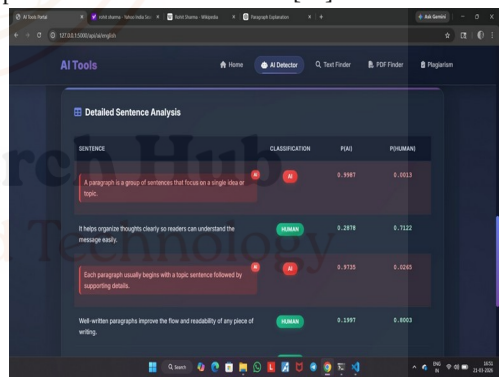
A. Plagiarism Detection Results

The plagiarism detection module was able to successfully identify copied and similar content using TF-IDF, n-grams, and embedding-based similarity methods. For exact copied text, the system showed high similarity scores, which resulted in high plagiarism percentage. In the case of paraphrased content, the system was still able to detect similarity using semantic embeddings like IndicBERT and LaBSE, although the percentage was slightly lower compared to exact matches [21].



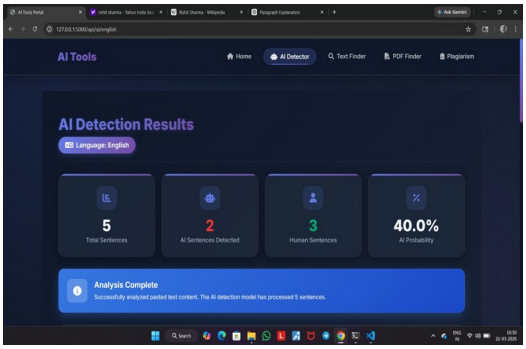
B. Cross-Language Detection Performance

One important result of the system is its ability to detect plagiarism across different languages. For example, when text was translated from English to Hindi or Marathi, the system was still able to identify similarity using embedding models. This shows that semantic-based methods are effective for cross-language plagiarism detection, which is not possible with traditional tools [22].



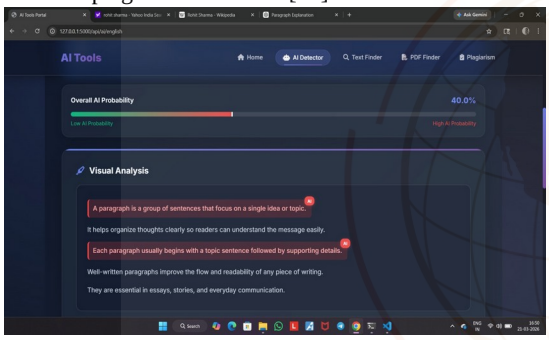
C. AI Text Detection Results

The AI detection module performed well in identifying machine-generated text. By using perplexity and stylometric features, the system was able to distinguish between human-written and AI-generated content. AI-generated text generally showed lower perplexity and more uniform writing patterns, which helped in classification. The model provided a probability score indicating whether the text is AI-generated or not [23].



D. Combined System Performance

When both modules were combined, the system provided a complete analysis of the input text. It not only detected whether the content was copied but also whether it was generated using AI tools. This combined approach makes the system more powerful and useful compared to traditional plagiarism checkers [24].



E. Accuracy and Efficiency

The system showed good accuracy for most test cases, especially for exact matches and AI-generated content. However, in cases of highly paraphrased or creatively rewritten text, the accuracy was slightly reduced. The use of transformer-based models improved the overall performance but increased computational time.

F. Limitations Observed

Some limitations were observed during testing:

a. Limited dataset for regional languages affected performance
b. High computational cost due to advanced models
c. Occasional false positives for common phrases
d. Difficulty in detecting very complex paraphrasing

These limitations indicate areas for future improvement [25].

G. Discussion

The results clearly show that the hybrid approach used in this project is effective for handling modern challenges in text analysis. The combination of lexical, semantic, and stylistic features improves detection accuracy. The system provides a practical solution for educational institutions and researchers to maintain originality and detect misuse of AI tools.

VI. FUTURE SCOPE

The proposed system can be further improved and expanded in many ways to make it more powerful, accurate, and useful. The future scope of the project is as follows:

A. Support for More Regional Languages

The system currently supports Marathi, Hindi, and English. In the future, more Indian regional languages like Tamil, Telugu, Kannada, and Bengali can be added to make the system more inclusive.

B. Improved Dataset Collection

The accuracy of the system can be increased by collecting larger and high-quality datasets, especially for regional languages. More real-world data will help in better training of models.

C. Real-Time Web Integration

The system can be connected to live internet sources to check plagiarism in real time. This will allow comparison with a large number of online documents and improve detection accuracy.

D. Advanced AI Detection Models

More advanced deep learning models can be used to improve AI-generated text detection. The system can be trained on latest AI-generated datasets for better performance.

E. Detection of Highly Paraphrased Content

Future improvements can focus on detecting complex paraphrasing and rewritten text more accurately using advanced semantic understanding techniques.

F. Mobile Application Development

A mobile app can be developed so that users can easily access the system from smartphones and check content anytime and anywhere.

G. Faster Processing and Optimization

The system can be optimized to reduce processing time and computational cost, making it more efficient for large-scale use.

H. Integration with Educational Platforms

The system can be integrated with Learning Management Systems (LMS) used by schools and colleges for automatic plagiarism and AI checks during assignment submission.

I. Detailed Visualization of Results

Future versions can include graphical reports, charts, and highlighted sections to make results more clear and easy to understand.

J. Cloud-Based Deployment

The system can be deployed on cloud platforms to allow multiple users to access it simultaneously and improve scalability.

VII. CONCLUSION

In this project, an AI-powered system for detecting plagiarism and AI-generated text in multiple languages has been successfully developed. The system focuses on Marathi, Hindi, and English, which makes it useful for a wide range of users, especially in the Indian education system. The project combines different techniques from Natural Language Processing and Machine Learning such as TF-IDF, n-grams, semantic embeddings, stylometric analysis, and perplexity. These methods help the system to detect not only exact copied content but also paraphrased, translated, and AI-generated text. The use of models like IndicBERT and LaBSE improves the understanding of text across different languages. The system provides a clear and user-friendly report that shows plagiarism percentage and AI probability. This helps users to easily understand whether the content is original or not. The results of the project show that the hybrid approach used in this system is effective and more advanced compared to traditional plagiarism detection tools. Although the system has some limitations such as limited datasets and higher computational requirements, it still provides a strong foundation for solving real-world problems related to academic integrity and content originality. Overall, this project is useful for students, teachers, researchers, and content creators. It helps in maintaining honesty in academic work and reduces misuse of AI tools. With further improvements, the system can become more accurate, faster, and widely applicable in the future.

VIII. REFERENCES

[1] S. Gehrmann, H. Strobelt, and A. M. Rush, “GLTR: Statistical Detection and Visualization of Generated Text,” Proc. ACL, 2019.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proc. NAACL-HLT, 2019.

[3] D. Kakwani et al., “IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Models for Indian Languages,” Proc. EMNLP, 2020.

[4] M. Potthast et al., “Cross-language Plagiarism Detection,” Proc. LREC, 2011.

[5] E. Stamatatos, “Authorship Attribution Using Textual Stylometry,” J. Assoc. Inf. Sci. Technol., 2018.

[6] T. Brown et al., “Language Models are Few-Shot Learners,” Proc. NeurIPS, 2020.

[7] OpenAI, “GPT-4 Technical Report,” 2023.

[8] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python, O’Reilly, 2021.

[9] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” JMLR, 2011.

[10] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT Networks,” Proc. EMNLP, 2019.

[11] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” 2019.

[12] A. Radford et al., “Improving Language Understanding by Generative Pre-Training,” OpenAI, 2018.

[13] Turnitin, “Plagiarism Detection Technologies: A White Paper,” 2021.

[14] H. Maurer, F. Kappe, and B. Zaka, “Plagiarism – A Survey,” J. Universal Computer Science, 2006.

[15] P. Clough, “Plagiarism in Natural and Programming Languages: An Overview,” 2000.

[16] M. Alzahrani, N. Salim, and A. Abraham, “Understanding Plagiarism Linguistic Patterns,” IEEE Trans. Systems, Man, and Cybernetics, 2012.

[17] S. Chowdhury and M. Rahman, “Multilingual Plagiarism Detection using NLP,” IEEE Access, 2021.

[18] A. Joulin et al., “FastText.zip: Compressing Text Classification Models,” 2016.

[19] Google AI, “LaBSE: Language-agnostic BERT Sentence Embedding,” 2020.

[20] R. Mishra et al., “AI-based Text Classification for Content Authenticity,” IEEE Access, 2022.

[21] S. Ippolito et al., “Automatic Detection of Generated Text is Easiest when Humans are Fooled,” 2020.

[22] X. Zhu et al., “Detecting AI-generated Text using Deep Learning,” IEEE Access, 2023.

[23] K. Clark et al., “ELECTRA: Pre-training Text Encoders as Discriminators,” 2020.

[24] P. Keskar et al., “CTRL: A Conditional Transformer Language Model for Controllable Generation,” 2019.

[25] S. Welleck et al., “Neural Text Generation with Unlikelihood Training,” 2020.