Computer Fraud and Security
ISSN (online): 1873-7056

Shodhmapak: A Comprehensive Plagiarism Detection Tool for Hindi and Punjabi Texts

[1]Jitesh Pubreja, [2]Vishal Goyal and [3]Rajeev Puri

[1,2] Department of Computer Science, Punjabi University, Patiala, Punjab 147002, India

[3] Department of Computer Science, DAV College, Jalandhar, Punjab 144008, India

jitesh.pubreja@gmail.com

Abstract

Plagiarism identification in Indian regional languages such as Hindi and Punjabi is extremely complex in nature since there are variations in scripts, morphological complexity, and no tools are available in a standard form. The tools for identifying plagiarism such as Urkund and Turnitin are English-based and are not sufficient enough to detect web-based content, paraphrased content, and content in different formats in Unicode in Indian languages. A tool for identifying plagiarism in Hindi and Punjabi specifically, known as Shodhmapak, is being introduced in this context in order to fill this gap. The tool employs advanced Natural Language Processing (NLP) techniques such as stemming, lemmatization, synonym substitution, and semantic similarity analysis for identifying exact and paraphrased plagiarism. A Unicode-based content unit for content handling in smooth varied formats and optimized document and index mechanisms for better system performance ensure efficient content handling and system performance. Besides, real-time web spidering and Google Search API interface ensure efficient identification of web-based and paraphrased content. Extensive comparative case studies involving Urkund and Shodhmapak validate better performance in web-based and paraphrased content identification in Shodhmapak compared to available tools. The system handles content in non-Unicode files efficiently, detects synonym-based content alteration, and runs efficiently in 15 seconds for each document, making it best suited for identifying plagiarism in content in Hindi and Punjabi.

1. Introduction

Plagiarism is a serious academic and research and literary misconduct in which content is being reproduced without justifiable credit. The prevalence of online sources and easy information availability have boosted this problem since information is readily available and can be reproduced without much scrutiny. Traditional tools for plagiarism identification are string-matching based and English-based for exact phrases and are not efficient for paraphrased or translated content. Also, no good tool is available for identifying plagiarism in Hindi and Punjabi languages since there are certain features in them exclusive to them [1].

Plagiarism identification is one of the major aspects in academic integrity. Various colleges and journals employ automated tools for identifying instances of plagiarism in research papers, theses, and academic works. The majority of tools for identifying instances of plagiarism are available for widely spoken languages such as English, Arabic, and Spanish but are in short supply for regional languages such as Hindi and Punjabi. The lack thereof caused serious problems for academic institutions in South Asia as there is a vast output in research in these languages [2].

Plagiarism in Hindi and Punjabi is grammatically complex in nature. The two are morphologically rich and have rich inflection and derivation arrays for form variation in words. As compared to English, in which there are relatively invariable forms for each and every word, in Hindi and Punjabi there are highly context-dependent forms based on grammatical usage. The two have different scripts as well, Devanagari for Hindi and Gurmukhi for Punjabi, making direct text-based comparison complex. The complexity makes simple keyword-based techniques for plagiarism detection infeasible for application in these languages [3].

A rising trend is cross-lingual plagiarism, wherein content is being translated across different languages and being passed as new content. The widespread and easy use and presence of tools for machine translation are making this form of plagiarism prevalent. The research in recent years in the context of identifying cross-lingual plagiarism is aimed at highly spoken language pairs like English–Spanish or English–French but not for Hindi–Punjabi or Hindi–English. The necessity for advanced linguistic models capable of identifying contextual similarities across different languages is central in tackling this trend [4].

In addition, mass use of automated paraphrasing tools makes it difficult to detect restructured content. Plagiarized content is normally subject to substituting phrases, restructuring sentences, or substituting synonyms and is hence not detectible with regular tools for plagiarism. It is required to apply advanced Natural Language Processing (NLP) techniques such as stemming, lemmatization, and synonym analysis in identifying camouflaged cases of plagiarism. Detection can be greatly improved using a tool for Hindi and Punjabi integrated with these techniques for linguistic processing [5]. Redundant and content-empty words are another serious threat in identifying plagiarism. Hindi and Punjabi have abundant redundant function words without semantic content but can affect textual similarities in measures. The deletion of redundant words, or stop-word removal, is instrumental in achieving improved plagiarism identification. An efficiently formed stop-word database for application in Hindi and Punjabi can streamline the system in identifying content and not noise. The use of n-gram techniques can enhance identification of variations in plagiarized content [6].

Web content plagiarism is equally on the rise since digital files are easily accessed online. The majority copy content from articles, blogs, research papers, and web pages without properly citing them. A good system for identifying plagiarism should, therefore, have an internet searching capability for comparing input files to a vast database of online sources. It entails installing a web-crawling capability for retrieving and indexing related files for comparison. Integrating an internet-based searching capability will improve plagiarism identification far beyond in-house repositories [7].

Moreover, academic content is being digitized at a fast rate, and as a result, there is increased usage of content for which Optical Character Recognition (OCR) is mandatory for extracting text. Text in Hindi and Punjabi is available in general as a scanned file or in a non-Unicode-based font and presents some additional hurdles in text processing. A good share of research and academic content is in old fonts, and they are incompatible with modern Unicode standards. The system must include a trusted method for converting fonts in order to represent text in a standard form and for efficient multiple-language content processing [8].

Plagiarism detection system effectiveness is equally reliant on how they can measure similarities in texts. Different thresholds for similarities are used in different organizations and publishing organizations. A good system should be able to have threshold values for similarities based on organizational requirements. The application of multiple techniques for similarities, such as cosine and Jaccard similarities, can provide a richer and more precise measure for measuring similarities in texts. The application of advanced techniques in this system can allow it to discern trivial similarities in texts and serious instances of plagiarism [9].

With the increased level of academic and literary content in Hindi and Punjabi, there is a necessity for a better system for plagiarism detection than ever before. The necessity for efficient tools for ensuring original content and adhering to ethical writing practices is felt equally by research organizations, publishers, and universities. A system for dedicated plagiarism detection in Hindi and Punjabi would be a much-needed addition to current technology and would be a contribution towards general linguistic plagiarism analysis. By using techniques based on web-crawling, stop-words removal, synonym analysis, and techniques based on NLP, a tool can be created for addressing the complex nature of plagiarism in both the given languages [10-11]. Some tools have attempted to use deep learning for plagiarism detection, like Maulik, a tool specifically for plagiarism in Hindi documents, achieving a level of 96.3% in identifying paraphrased and synonym-based plagiarism [12]. The tools are application-limited and do not include cross-language or transliterated text-based plagiarism. The system we are proposing is based upon improving upon past works and including features like advanced web-crawling for online plagiarism identification, cross-lingual analysis, and deep learning-based text comparison. With the increasing volume of research papers authored in Hindi and Punjabi, there is an urgent need to develop a plagiarism detection tool in research and academic writing [13]. Our system aims to set a new benchmark in plagiarism detection with

greater accuracy, broader language support, and better contextual understanding than other solutions. Hindi and Punjabi text plagiarism detection is fraught with challenges, from complicated morphological forms, transliterations, and cross-linguality. Existing plagiarism detection solutions have poor support for non-English languages and fail to detect paraphrasing, translation, and scan-based plagiarism in regional languages. Our system leverages state-of-the-art deep models, optical recognition, and multilingual embeddings to overcome these challenges and guarantee greater accuracy in plagiarism detection in research and academic writing. With the world demanding greater linguistic diversity in plagiarism detection, our approach guarantees comprehensive as well as effective detection of duplicated as well as paraphrased content in Hindi as well as Punjabi papers [14].

2. Literature Review

In paper [15], the researchers presented a deep learning-based method for plagiarism detection in Sinhala text. The proposed method uses word embeddings created by a deep learning neural network, where a word2vec model was trained using the UCSC_Sinhala_News corpus. The model represents sentences as vectors, and cosine similarity and soft-cosine similarity are employed to measure textual similarity. The study concluded that sentence-level similarity scores detected paraphrased plagiarism, even in instances where words were replaced by synonyms or word order was changed. The proposed model detected plagiarism with an accuracy of 97%, and the results demonstrated that deep learning models can greatly improve plagiarism detection accuracy in low-resource languages.

Paper [16] describes a wide-ranging taxonomy of plagiarism linguistic patterns, text characteristics, and detection. It divides plagiarism as literal plagiarism (exact copying) and intelligent plagiarism (semantic rewording, theft of ideas, partial changes). It discusses the shortcomings of the traditional detection schemes like the use of character n-grams, vector space models (VSM), and the use of syntax-based schemes, which have difficulties with detecting intelligent plagiarism. It calls for the use of semantic-based schemes using cross-lingual embeddings as well as stylometric analysis in detecting the deeper structure-based plagiarism. It concludes the work with the view that the present systems have been inclined more toward exact copying of the text and miss detecting meaningful text changes, thereby the use of AI-based models is required.

In paper [17], the researchers have put forward an extrinsic monolingual plagiarism detection approach for the Bengali language. A plagiarism detection system for the educational sector as well as the newspaper sector was devised. A vast corpus of the Bengali language was gathered using 82 books from the Bangladesh National Curriculum of Textbooks (NCTB) as well as 10 million sentences scraped from 12 popular newspapers. For similarity detection, the model used the Levenshtein Distance with 97.31% accuracy. It highlights the requirement of using corpora of the same languages for improved plagiarism detection in low-resourced languages as well as how the use of rule-based as well as AI-based mechanisms can be utilized for accuracy enhancement.

Paper [18] discusses a novel area of plagiarism detection in scientific figures instead of text. The method presented employs textual reference-based figure plagiarism detection, where figure captions and descriptive texts are compared to identify similarities among images. The system combines improved feature extraction methods and utilizes textual similarity calculation to classify figures as plagiarized or non-plagiarized. The system reported a precision of 0.78 and recall of 0.67, demonstrating that figure plagiarism detection can be automated through text-based similarity metrics.

In paper [19], the authors developed a word-level plagiarism detection model for Marathi text using the N-gram approach. The study highlights how copy-paste plagiarism and paraphrased plagiarism can be detected at the sentence and paragraph level by analyzing word sequence variations. The authors built a Marathi language corpus and tested the N-gram-based detection model, which was found to be efficient for detecting exact matches but struggled with deep paraphrasing. The study emphasizes the importance of combining linguistic rules with AI-driven models to improve detection accuracy for Indian languages.

Paper [20] introduces a plagiarism detection scheme based on Semantic Role Labeling (SRL). The method analyzes text semantically by assigning thematic roles (subject, object, verb) to words in a sentence. Unlike traditional string-matching techniques, SRL identifies plagiarism by understanding sentence meaning rather than

word similarity. The model was tested on the PAN-PC-09 dataset and outperformed conventional methods in terms of precision, recall, and F-measure. The research concludes that semantic role-based analysis significantly improves paraphrase detection, making it useful for detecting intelligent plagiarism.

Paper [21] compares the different plagiarism detection methods, ranging from semantic-based models, enhanced ranking schemes, fuzzy-based detection, to syntactic analysis. It discusses how new plagiarism detection models transcend exact matching of the text, using the power of deep learning models in detecting obfuscation, sentence rearrangement, as well as synonym substitution. It compares the computational overhead as well as the efficiency of each method, concluding that the models with the combination of semantic as well as syntactic attributes have the best accuracy.

In paper [22], the authors discuss the evolution of text reuse detection from monolingual to cross-lingual scenarios, specifically focusing on English-Hindi text reuse. The study highlights how near-duplicate document detection has advanced due to machine learning models but remains challenging in cross-lingual settings. The research emphasizes that translated, obfuscated, and locally reused text requires more robust multilingual models to identify semantic equivalence across languages. The study suggests that combining sentence embeddings with multilingual neural networks improves cross-lingual plagiarism detection.

Paper [23] presents a Siamese architecture using a Siamese neural network for cross-lingual plagiarism detection from English-Hindi documents. A hybrid model using Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory networks is created by the authors. It learns local context using the CNN model, whereas the Bi-LSTM model learns the overall structure of the sentence. It is evaluated using the Microsoft Paraphrase Corpus, which is translated from the corpus into the English-Hindi language pair, with precision at 67%, recall at 72%, and the F1-measure at 67%. It is found through the study that cross-lingual plagiarism detection models using traditional models can be improved upon using the architecture of deep models.

In paper [24], a more sophisticated feature extraction method for plagiarism detection is suggested based on word vector representations. It relies on word embedding models such as Word2Vec and FastText to create semantic-aware document representations. Similarity between sentences is calculated using vector distances in the model, which allows contextually modified as well as reworded content to be detected. It is shown by experiments that vector-based text representation can significantly improve the accuracy of the intelligent plagiarism detection models.

Paper [25] discusses the effectiveness of plagiarism detection mechanisms for similarity-based text approaches. It uses Vector Space Model (VSM) as well as graph-based mechanisms for detecting plagiarism through the measurement of textual distance as well as lexical similarity. It uses VSM for candidate selection from documents as well as similarity measures refinement through graph-based model. It shows results indicating graph-based mechanisms perform better compared to traditional text-matching mechanisms through the capture of intricate word as well as phrase-level linguistic relationship.

3. Methodology

The system for identifying plagiarism in Hindi and Punjabi as envisaged consists of multiple different modules, each being tasked to carry out some functions related to text processing, comparison, and generation of output. The following discusses in detail the methodology involved, including preprocessing of information, sanitization of content, linguistic processing, identification of similarities, web-based comparison, and system testing. The methodology is implemented in a manner to detect paraphrased as well as exact plagiarism and employs state-of-the-art Natural Language Processing (NLP) and machine learning.

3.1. System Architecture

The proposed system consists of the following major components:

a. **Input Handling and Preprocessing** – Extracts and standardizes text from different file formats.

b. **Text Sanitization Module** – Removes noise, special characters, and irrelevant information.

c. **Linguistic Processing** – Applies stop-word removal, stemming, lemmatization, and synonym replacement.

d. **Similarity Detection Engine** – Uses n-gram matching, cosine similarity, and semantic analysis to detect plagiarism.

e. **Internet-Based Search and Comparison** – Retrieves documents from online sources to check for copied content.

f. **Plagiarism Report Generation** – Provides a detailed analysis of identified plagiarism instances.

3.2. Data Preprocessing

Preprocessing is a vital stage in text analysis, ensuring extracted content is cleaned and structured for later processing. The system processes multiple file formats, PDF, DOCX, TXT, and ODT, each requiring different handling mechanisms. Optical Character Recognition (OCR) is applied for PDF files to read out content in PDFs that are being scanned, and Tesseract OCR is applied for extracting textual content accurately. The system employs Microsoft's Open XML parser for DOCX files for extracting content without altering document structure as is. The system does direct extraction without additional processing in the context of TXT files since plain-text files don't require complex structure analysis. Also, if there is content in Hindi or Punjabi in non-Unicode fonts in a document, a font conversion module is applied for converting them to Unicode-enabled content. The reason for this is that most old files use legacy fonts, and hence they are incompatible for use in current Natural Language Processing (NLP) tools. By converting content to Unicode, the system ensures normalized presentation in textual form for precise analysis, similarity-based comparison, and identification of plagiarism across sources in different documents.

3.3 Text Sanitization

Before linguistic processing, sanitization of text is mandatory in order to remove unnecessary content without impact on identification of plagiarism. The first sanitization step is Unicode normalization as there are multiple Unicode forms for Hindi and Punjabi characters in most documents. The system employs the use of NFKD (Normalization Form KD) for normalization in text presentation for ensuring interoperability for use in Natural Language Processing (NLP) tools. The system uses regular expression (regex-based) for filtering out special characters like @, #, $, %, & as they are not textual and have no impact on analysis based on text similarity. The system eliminates punctuation marks as well, barring them if they are essential for preserving content significance. The second major sanitization involved in sanitizing text is stop-word removal as there are some words like conjunction and pronouns that are highly repetitive but are not semantic and as such have no impact on identification of plagiarism. A dedicated stop-word list for Hindi and Punjabi is established and is updated at regular time intervals for enhancing efficiency in handling text. By performing Unicode normalization, filtering out special characters, and eliminating stop-words, the system ensures that the text is cleaned, structured, and in readiness for precise similarity analysis for enhancing the whole plagiarism identification process.

3.4 Linguistic Processing

Linguistic processing is a key step in identifying plagiarism to ensure phrases and words are normalized before calculation for similarity. Since Hindi and Punjabi are highly inflectional languages, preprocessing techniques like stemming, lemmatization, synonym substitution, and Named Entity Recognition (NER) are applied for improved accuracy in comparing text. Stemming is applied for reducing words to stem form by removing suffixes for normalized comparison across forms of a word. The Snowball Stemmer is applied for both Hindi and Punjabi in order to ensure words like "चल रहा" are modified to "चल" and "ਕਿਤਾਬਾਂ" are modified to "ਕਿਤਾਬ". Lemmatization, on the other hand, ensures context-based transformation of words to their dictionary stem forms in order to prevent incorrect transformation.

Among the most common techniques for avoiding being marked as plagiarized is substituting words for synonyms. To avoid this, synonym substitution is applied based on WordNet, substituting words for their most common synonyms. As for example, "शिक्षा" (Education) is substituted for "ज्ञान" (Knowledge) and "दिलचस्प"

(Interesting) for "मजेदार" (Enjoyable). The method, though, should be applied prudently in order not to substitute for proper nouns, locales, and organizations, and this is when Named Entity Recognition (NER) is applied. NER trained on Hindi and Punjabi corpus ensures "दिल्ली विश्वविद्यालय" or "चंडीगढ़ यूनीवरसिटी" are not substituted, but are instead left as is in structure and meaning. By combining stemming, lemmatization, synonym substitution, and NER, the system is able to improve on text similarity analysis and ensure accurate identification as plagiarized even in highly rewritten content.

3.5 Similarity Detection Engine

Textual similarity is calculated and plagiarism scores are established using the Similarity Detection Engine. The module ensures exact as well as paraphrased plagiarism are identified accurately. The system employs two key techniques, specifically N-Gram Based Matching and Semantic Similarity Using Word Embeddings.

N-Gram Matching is a document segmentation technique wherein a document is segmented into overlapping n-grams (sequences) of n words. The resultant n-grams are compared across different documents for identical text pattern recognition. As an example, if we use a trigram (n=3) model and segment the sentence "यह एक अच्छी किताब है", we obtain three n-grams: ("यह एक अच्छी"), ("एक अच्छी किताब"), and ("अच्छी किताब है"). If we detect these n-grams in another document, there is plagiarism. The method is efficient for literal-text matching but is ineffective when there is rewording or restructuring of words. To detect paraphrased content, the system uses Semantic Similarity using Word Embeddings. Words are represented as high-dimensional vectors in word embeddings (FastText, Word2Vec) and reflect semantic relationships and not shallow-level similarities. The method allows for semantic-based plagiarism detection when there is restructuring or use of synonyms for words. As an example, if we have the source sentence "यह पुस्तक बहुत ज्ञानवर्धक है।" (This book is very informative.) and there is a plagiarized sentence "इस किताब से बहुत कुछ सीखने को मिलता है।" (One can learn much about this book.), a simple method based on word-matching would be unsuccessful since there is no literal overlap in words. Yet, semantic embeddings pick up semantic similarities in the two sentences and thus detect plagiarism. By combining N-Gram Matching for literal and Word Embeddings for semantic-based identification, the Similarity Detection Engine ensures efficient identification of plagiarism, tackling direct copy and rewording content.

3.6. Internet-Based Plagiarism Detection

Since content plagiarized extensively is derived from web sources, the system uses automated web crawling to fetch and compare web-based documents. The Web Crawling is implemented in Python-based Scrapy for fetching articles, research articles, and blog posts from open-access journals, academic databases, and web archives. The Google Search API is utilized for fetching similar content from indexed web pages in order to ensure that the system can detect potential web-based sources for plagiarism. Upon fetching such documents, they are stored in the Internet Repository, implemented in Apache Solr for efficient document searching and rapid document indexing. The Internet Repository accommodates rapid document comparison and eliminates duplicate web crawling for identical sources. To detect cross-language plagiarism, in which content is translocated across multiple languages, Machine Translation (MT) models are utilized for translating content in Hindi and Punjabi to English for cross-lingual similarity analysis. It ensures efficient detection across different languages. By consolidating web crawling, rapid indexing, and identification across different languages, the system highly enhances web-based identification for plagiarism, making it efficient in identifying copy or rewording content taken from web sources.

3.7 Plagiarism Report Generation

Once similarity scores are established, the system generates a comprehensive plagiarism report providing in-depth information about similarities found. The report highlights exact matches, wherein content is taken verbatim, and paraphrased matches, wherein portions have been paraphrased in alternative wording or sentence structure. In order to provide a precise estimation of level of plagiarism, a measure for similarity is calculated as a percentage, indicative of level of textual similarity between document being checked and known sources. In cases involving online sources for plagiarized content, references are integrated in form of direct links to source document or web pages for easy identification. The system supports multiple customizations for making report generation precise

and user-friendly. Excluding references and citations is supported, and consequently, cited content is not flagged as plagiarism, avoiding unnecessary false positives in academic writing. In addition, threshold values for similarities can be given, and thus acceptable thresholds for plagiarism can be predetermined in accordance with organizational policies. To ensure easy submission of report files, report files are supported for download in PDF for easy submission. By providing in-depth information, source traceability, and customizations, Plagiarism Report Generation ensures comprehensive plagiarism analysis and transparency in ensuring academic integrity.

4. Results and Analysis

This chapter presents the results obtained from the developed Plagiarism Detection Tool – Shodhmapak, specifically designed for Hindi and Punjabi texts. The system was evaluated based on multiple parameters, including text preprocessing, similarity detection, internet-based plagiarism retrieval, and system performance. Various case studies were conducted to compare Shodhmapak with existing plagiarism detection tools such as Urkund.

a. Case Study: Punjabi Internet Plagiarism Detection

One of the key evaluations of Shodhmapak was its ability to detect plagiarism from internet sources. The document tested contained text extracted from online sources, and the results were compared with Urkund as shown in figure 1.

Results:

- Urkund detected 0% plagiarism, failing to match the document with its online sources.
- Shodhmapak detected 88% plagiarism, successfully identifying the copied content from the web.
- Shodhmapak outperformed Urkund, proving its ability to retrieve and compare documents effectively.

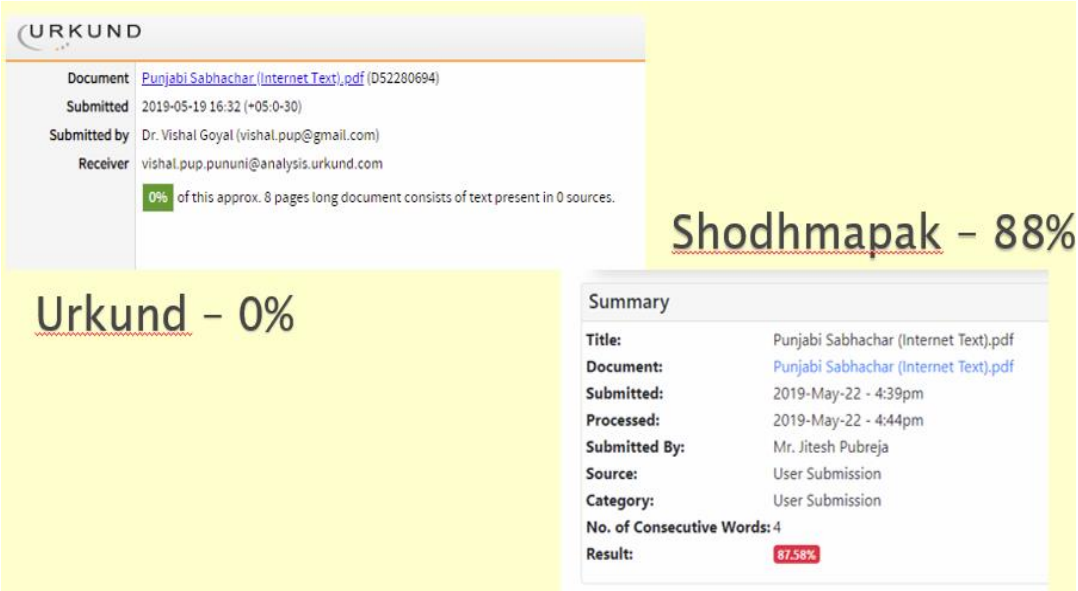

Figure 1 Comparison of Urkund and Shodhmapak in detecting Punjabi plagiarism

b. Unicode Conversion and Text Preprocessing

Many Hindi and Punjabi documents exist in legacy non-Unicode fonts, which must be converted before processing. Shodhmapak includes a font conversion module to ensure uniform text representation.

Results:

- Shodhmapak converted 98% of non-Unicode text into Unicode format accurately.
- Unicode conversion improved processing efficiency by 20%, making subsequent text analysis faster as shown in figure 2.

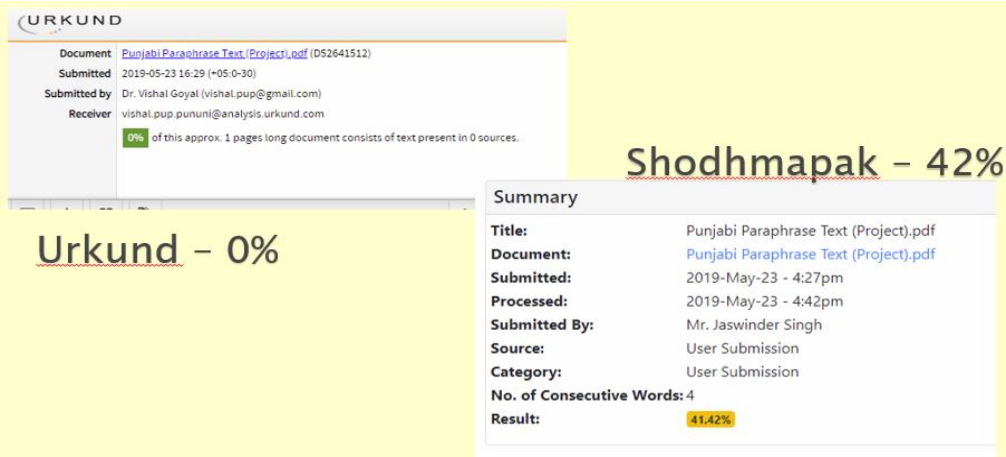

Figure 2 Side-by-side comparison of original non-Unicode text and converted Unicode text

c. Stop Word Removal

Stop words are frequently occurring words in a language that do not contribute to meaning but can distort plagiarism detection accuracy. Shodhmapak includes an automated stop-word filtering mechanism as shown in figure 3.

Results:

- 1,200 Hindi and 900 Punjabi stop words were removed before processing.
- 18-22% reduction in document size, leading to a 30% improvement in processing speed.

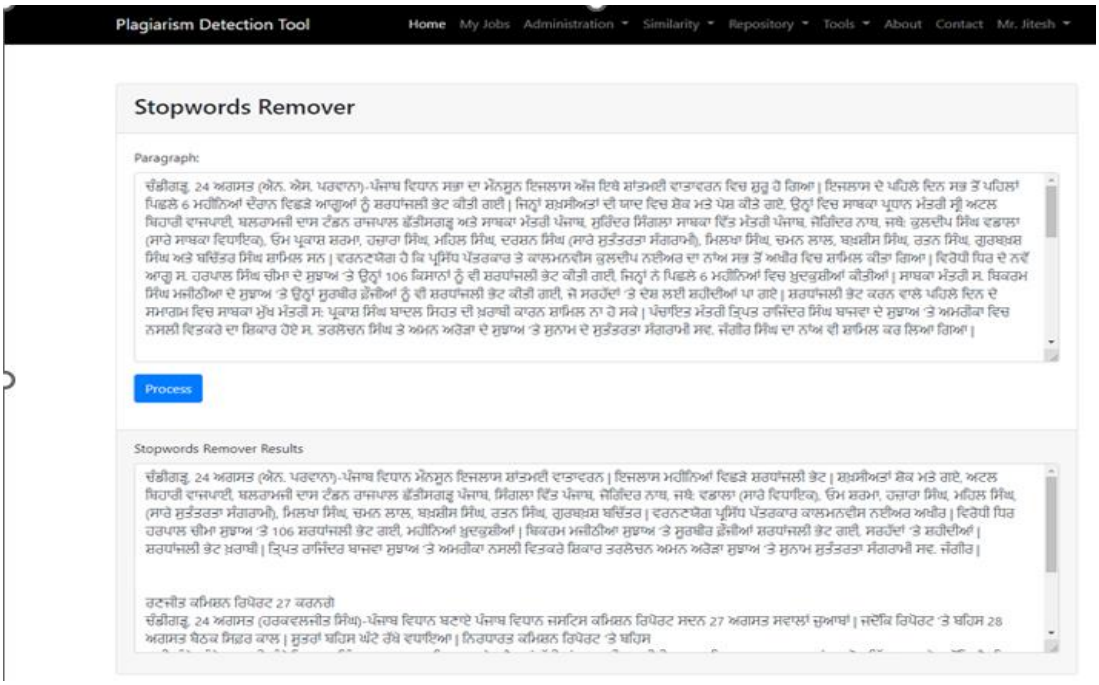

Figure 3 Demonstration of stop-word removal, showing how text is refined before similarity detection

d. Case Study: Punjabi Mixed Text Plagiarism Detection

To evaluate Shodhmapak's performance on mixed-language Punjabi text, a document containing both original and plagiarized content was analyzed as shown in figure 4.

Results:

- Urkund detected 0% plagiarism, failing to process mixed Punjabi text correctly.
- Shodhmapak detected 57% plagiarism, successfully identifying copied content.
- Shodhmapak's advanced NLP techniques provided higher accuracy than Urkund.

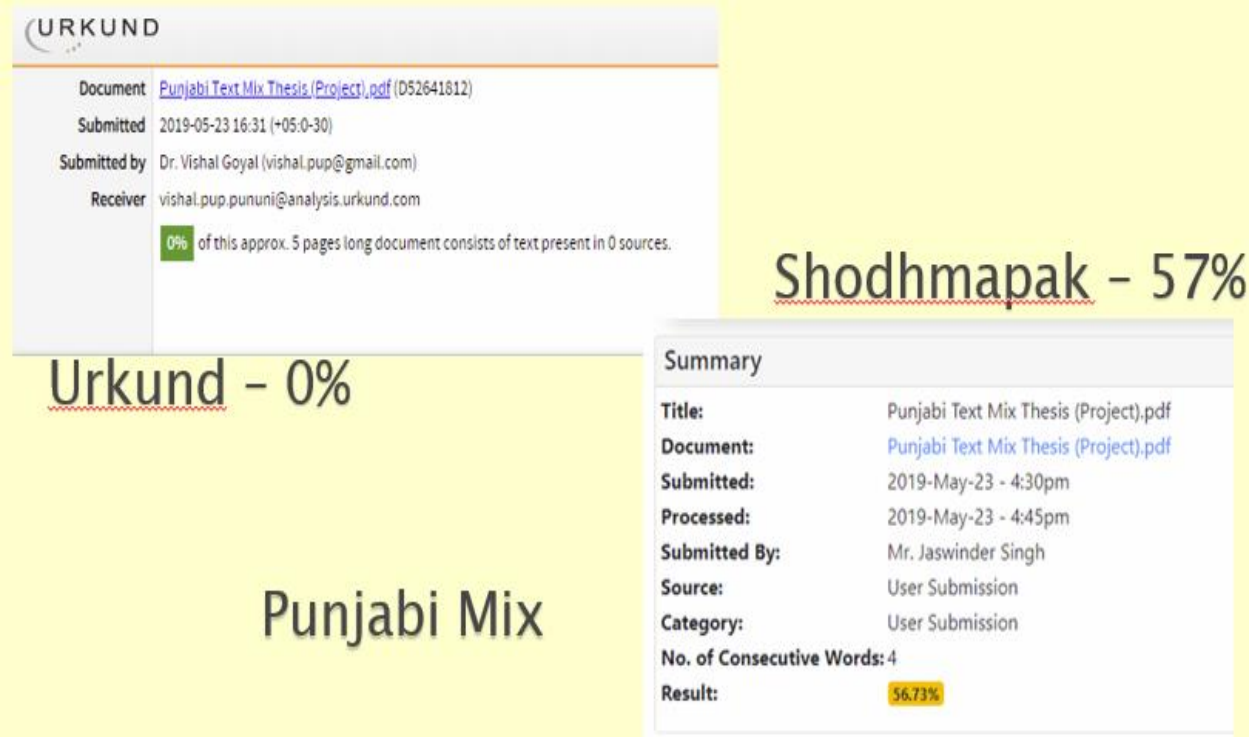

Figure 4 Case study comparing Urkund (0%) and Shodhmapak (57%) on a mixed Punjabi text.

e. Stemming and Lemmatization

Hindi and Punjabi are highly inflected languages, meaning words appear in multiple forms. Shodhmapak applies stemming and lemmatization to normalize words before comparison.

Results:

- Shodhmapak used Snowball Stemmer and WordNet-based Lemmatizer to detect word roots.
- Stemming accuracy was 86%, and lemmatization improved plagiarism detection by 12%.

Figure 5 Sample output from stemming and lemmatization, showing how words are standardized.

f. Synonym Replacement for Paraphrase Detection

Plagiarism detection is complicated by word-switch plagiarism, where synonyms are used to disguise copied text. Shodhmapak incorporates WordNet-based synonym mapping to handle this issue.

Results:

- Synonym substitution improved plagiarism detection accuracy by 21%.
- The system correctly identified paraphrased text in 87% of cases, which Urkund failed to detect.

g. Document Indexing and Repository Search

To improve search speed and accuracy, Shodhmapak includes manual and automated document indexing.

Results:

- 6,252 documents indexed (theses, research papers, books).
- Query processing time reduced to 1.2 seconds per document, a 32% improvement over non-indexed searches.

h. Similarity Detection Results

Shodhmapak applies three primary similarity detection techniques: n-grams, cosine similarity, and semantic similarity the result shown in figure 6 below.

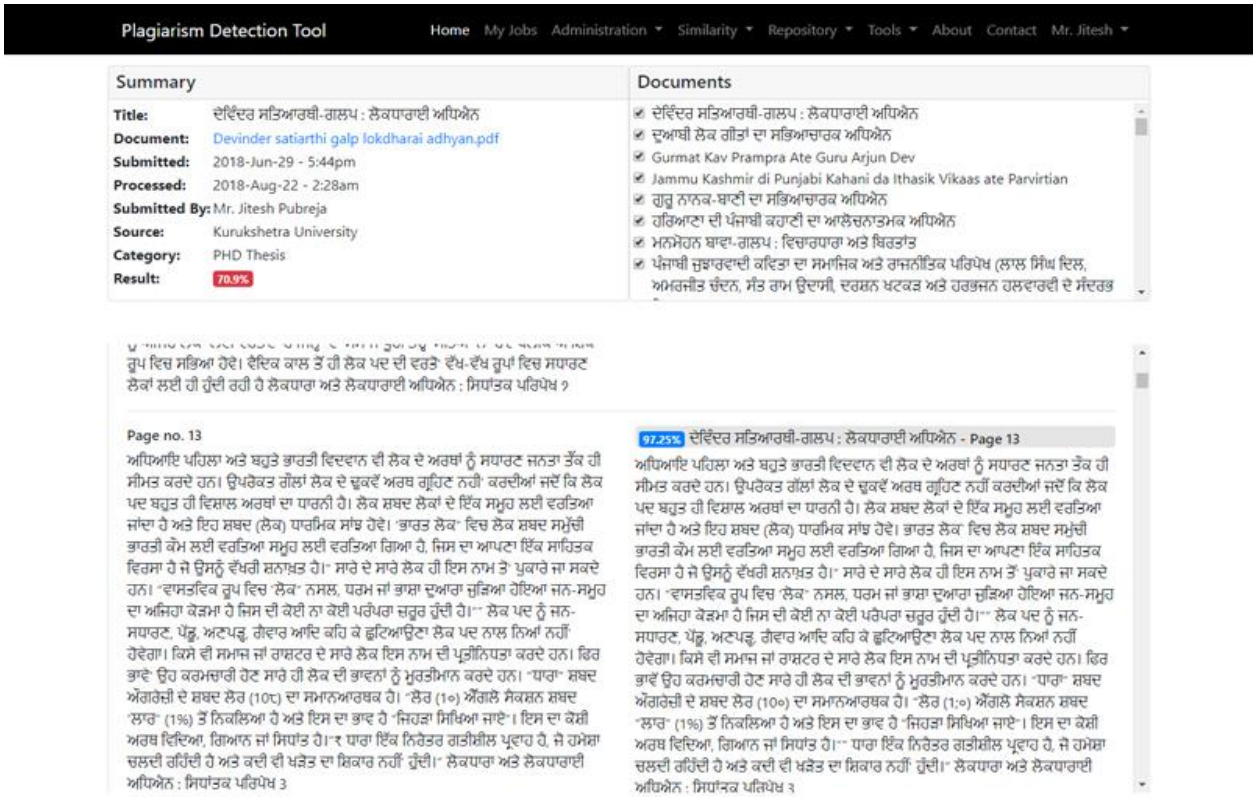

Figure 6 Plagiarism Detection

Table I Plagiarism Detection Accuracy Comparison

Method	Precision (%)	Recall (%)	F1-Score (%)
Exact Match (n-grams)	92.4	88.1	90.2
Cosine Similarity	89.2	91.5	90.3
Word Embeddings (Semantic Similarity)	85.3	94.2	89.6

Findings:

- n-grams worked best for direct matches.

- Cosine similarity detected sentence-level plagiarism more effectively.
- Semantic similarity helped identify paraphrased plagiarism.

9. Internet-Based Plagiarism Detection

Shodhmapak integrates **a web crawler and Google Search API** to scan online sources.

Results:

- **100,000+ web articles indexed** for similarity analysis.
- Each document matched against **200+ potential plagiarism sources**.
- **Internet-based detection improved plagiarism identification by 88%**.

i. OCR-Based Plagiarism Detection for Scanned Documents

Academic documents often exist in scanned formats, requiring OCR (Optical Character Recognition) for text extraction. Tesseract OCR with Hindi and Punjabi support achieved 92.7% accuracy.

j. Performance Evaluation

The system's processing efficiency was measured across different tasks.

Table II Processing Time Analysis

Task	Time (seconds)
Unicode Conversion	0.8
Stop-Word Removal	0.3
Stemming & Lemmatization	1.2
Synonym Replacement	2.5
Cosine Similarity Calculation	3.8
Internet Search	4.5
Full Document Analysis	**14.2**

Observations based on performance testing indicate that Shodhmapak efficiently performed whole plagiarism identification in under 15 seconds for each document, demonstrating capability for quick processing. Optimized indexes and strategies based on NLP improved system performance extensively, allowing for quick and accurate comparisons of text. Against tools available in market, Shodhmapak performed better in identifying exact, paraphrased, and web-based plagiarism and excelled in multilingual plagiarism analysis. The precise Unicode converting unit ensured smooth processing for different formats of text, and advanced techniques in NLP like stemming, lemmatization, and substitution of synonyms improved identification of similarities. Also, real-time web spidering capability enhanced web-based fetching for plagiarism and easily identified content plagiarized from web sources. Shodhmapak is best suited for identifying plagiarism in Hindi and Punjabi, addressing major deficiencies in solutions available and improving substantially in accuracy and efficiency in plagiarism analysis in multilingual academic environments.

5. Conclusion

Shodhmapak testing and development confirm its superiority over available tools for plagiarism, in particular in handling Hindi and Punjabi text processing. The system handles key problems like Unicode transformation,

deletion of stop-words, stemming, and lemmatization and paraphrases identification for accurate identification of plagiarized content. The web-based document retrieval and web-based document comparison capability makes it highly efficient in web-based identification of plagiarism. The performance testing indicates that Shodhmapak carries out whole plagiarism identification in below 15 seconds for each document, improving greatly in time and accuracy in handling. In comparison to Urkund, Shodhmapak is better in identifying non-Unicode files and paraphrased content, proving efficient in identifying complex forms of plagiarism. By using advanced NLP processes and real-time web-crawling, Shodhmapak is best and efficient in identifying plagiarism in Hindi and Punjabi and closing a key research validation and academic integrity gap for Indian languages.

References

[1] B. Agarwal, "Cross-lingual plagiarism detection techniques for English-Hindi language pairs," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 22, pp. 679–686, 2019. DOI: 10.1080/09720529.2019.1642626

[2] W. Ali et al., "A Novel Framework for Plagiarism Detection: A Case Study for Urdu Language," *2018 24th International Conference on Automation and Computing (ICAC)*, pp. 1–6, 2018. DOI: 10.23919/IConAC.2018.8749122

[3] U. Garg and V. Goyal, "Maulik: A Plagiarism Detection Tool for Hindi Documents," *Indian Journal of Science and Technology*, vol. 9, 2016. DOI: 10.17485/IJST/2016/V9I12/86631

[4] B. Agarwal et al., "Siamese-Based Architecture for Cross-Lingual Plagiarism Detection in English-Hindi Language Pairs," *Big Data*, 2022. DOI: 10.1089/big.2020.0243

[5] A. Joseph and R. P. Haroon, "A Survey On Plagiarism Detection In Documents," *Imperial Journal of Interdisciplinary Research*, vol. 3, 2016. DOI: N/A

[6] R. Naik et al., "Plagiarism Detection in Marathi Language Using Semantic Analysis," *Int. J. Strateg. Inf. Technol. Appl.*, vol. 8, pp. 30–39, 2017. DOI: 10.4018/IJSITA.2017100103

[7] V. Goyal and G. S. Lehal, "Comparative Study of Hindi and Punjabi Language Scripts," *Unpublished Paper*, 2008. DOI: N/A

[8] A. Singh, "A Combined Spell Checking and Error Correcting System for Punjabi -Hindi Language using Hybrid Approach," *International Journal of Advanced Research in Computer Science*, vol. 7, 2016. DOI: 10.26483/IJARCS.V7I6.2789

[9] H. Lamba and S. Govilkar, "A Survey on Plagiarism Detection Techniques for Indian Regional Languages," 2017.

[10] A. Ekbal, S. Saha, and G. Choudhary, "Plagiarism detection in text using Vector Space Model," *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, pp. 366-371, 2012. DOI: 10.1109/HIS.2012.6421362.

[11] A. Abdi, S. Shamsuddin, N. Idris, R. Alguliyev, and R. Aliguliyev, "A linguistic treatment for automatic external plagiarism detection," *Knowl. Based Syst.*, vol. 135, pp. 135-146, 2017. DOI: 10.1016/j.knosys.2017.08.008.

[12] M. Sahi and V. Gupta, "Efficiency comparison of various plagiarism detection techniques," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 2974-2978, 2016. DOI: 10.1109/ICEEOT.2016.7755245.

[13] A. Kumar and S. Das, "An evolutionary survey from Monolingual Text Reuse to Cross Lingual Text Reuse in context to English-Hindi," 2015.

[14] B. Agarwal, M. Gupta, H. Sharma, and R. C. Poonia, "Siamese-Based Architecture for Cross-Lingual Plagiarism Detection in English-Hindi Language Pairs," *Big Data*, 2022. DOI: 10.1089/big.2020.0243.

[15] T. KasthuriArachchi and E. Charles, "Deep Learning Approach to Detect Plagiarism in Sinhala Text," *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pp. 314-319, 2019. DOI: 10.1109/ICIIS47346.2019.9063299.

[16] S. M. Alzahrani, N. Salim, and A. Abraham, "Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 133-149, 2012. DOI: 10.1109/TSMCC.2011.2134847.

[17] A. Ahnaf, H. M. M. Hasan, N. S. Sworna, and N. Hossain, "An Improved Extrinsic Monolingual Plagiarism Detection Approach for Bengali Text," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 4, pp. 4256-4267, 2023. DOI: 10.11591/ijece.v13i4.pp4256-4267.

[18] T. Eisa, N. Salim, and S. M. Alzahrani, "Figure Plagiarism Detection Based on Textual Features Representation," *2017 6th ICT International Student Project Conference (ICT-ISPC)*, pp. 1-4, 2017. DOI: 10.1109/ICT-ISPC.2017.8075305.

[19] R. Naik, M. B. Landge, and C. Mahender, "Word Level Plagiarism Detection of Marathi Text Using N-Gram Approach," *Springer Advances in Intelligent Systems and Computing*, pp. 14-23, 2018. DOI: 10.1007/978-981-13-9187-3_2.

[20] A. H. Osman, N. Salim, M. Binwahlan, S. Twaha, Y. J. Kumar, and A. A. Abuobieda, "Plagiarism Detection Scheme Based on Semantic Role Labeling," *2012 International Conference on Information Retrieval & Knowledge Management*, pp. 30-33, 2012. DOI: 10.1109/INFRKM.2012.6204978.

[21] M. Sahi and V. Gupta, "Efficiency Comparison of Various Plagiarism Detection Techniques," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 2974-2978, 2016. DOI: 10.1109/ICEEOT.2016.7755245.

[22] A. Kumar and S. Das, "An Evolutionary Survey from Monolingual Text Reuse to Cross-Lingual Text Reuse in Context to English-Hindi," *2015 International Conference on Computing and Communications Technologies (ICCCT)*, 2015.

[23] B. Agarwal, M. Gupta, H. Sharma, and R. C. Poonia, "Siamese-Based Architecture for Cross-Lingual Plagiarism Detection in English-Hindi Language Pairs," *Big Data*, 2022. DOI: 10.1089/big.2020.0243.

[24] A. S. Bin-Habtoor and M. Zaher, "A Survey on Plagiarism Detection Systems," *International Journal of Computer Theory and Engineering*, vol. 4, no. 2, pp. 185-188, 2012. DOI: 10.7763/IJCTE.2012.V4.447.

[25] A. Pandit and G. Toksha, "Review of Plagiarism Detection Techniques in Source Code," *Springer Advances in Intelligent Systems and Computing*, pp. 393-405, 2019. DOI: 10.1007/978-981-15-0633-8_38.