Open Access Research Journal of
Engineering and Technology
Journals home page: https://oarjpublication/journals/oarjet/
ISSN: 2783-0128 (Online)

OARJ OPEN ACCESS RESEARCH JOURNALS

(RESEARCH ARTICLE)

A COMPUTATIONAL APPROACH TO PLAGIARISM DETECTION IN KANNADA TEXTS USING SENTENCE, BIGRAM, AND TRIGRAM SIMILARITY MODELS

U B Pavanaja * and Prajna Devadiga

Vishvakannada Foundation, Bengaluru, Karnataka, India.
* Corresponding Author

ORCID Details
U B Pavanaja: https://orcid.org/my-orcid?orcid=0009-0000-2148-9536
Prajna Devadiga: https://orcid.org/0009-0005-7627-2234

Open Access Research Journal of Engineering and Technology, 2026, 11(01), 077–084

Publication history: Received on 10 July 2026; revised on 16 August 2026; accepted on 18 August 2026

Article DOI: https://doi.org/10.53022/oarjet.2026.11.1.0060

ABSTRACT

This paper documents a Python implementation for comparing Kannada Unicode texts using sentence-level sequence similarity and character- and word-level bigram and trigram similarities. The application normalizes text with NFKC (Normalization Form Compatibility Composition), collapses whitespace, segments sentences using Kannada danda and common punctuation, computes Jaccard and cosine n-gram scores, and produces an interpretable report. In the supplied combined-mode run, *Kadyanata* contained 1,388 segmented source sentences and *Karavaliya Saviradondu Daivagalu* contained 219 target sentences. At a sentence threshold of 0.60, 138 target sentences matched their best source sentence, giving a sentence-match rate of 63.01%. Character bigram and trigram cosine similarities were 96.05% and 88.89%, while word bigram and trigram cosine similarities were 15.43% and 6.25%. The mean of the eight implemented n-gram scores was 36.08%. The code-defined combined score, 40% sentence-match rate plus 60% mean n-gram similarity, was 46.85%.

Keywords: Kannada Text Processing; Plagiarism Screening; Sentence Similarity; Bigram; Trigram; Jaccard Similarity; Cosine Similarity.

1 INTRODUCTION

Text-similarity screening assists human review by exposing shared passages, phrases, and document-level overlap. N-gram and vector-based measures are established lexical approaches to such comparison [1]. Kannada processing also requires attention to morphological variation and Unicode representation [2]. This study documents the actual operation of a lightweight Kannada similarity application rather than claiming a new algorithm or validated detection performance.

Automated plagiarism detection can be broadly divided into extrinsic and intrinsic approaches. Extrinsic detection compares a suspicious document with one or more candidate source documents, whereas intrinsic detection identifies possible style inconsistencies within a single document. Since the present study directly compares a target Kannada

document with a supplied source document, it follows an extrinsic text-similarity setting [9]. However, lexical similarity measures are more effective for identifying direct or near-direct reuse than meaning-preserving paraphrases involving synonym substitution, structural reorganization, or semantic reformulation [10].

2 LITERATURE REVIEW

Kannada work has applied trigrams to paraphrase generation [3], rule-based lemmatization to inflectional forms [2], language models to downstream Kannada tasks [4], and lexical similarity plus classifiers to paraphrase-pair classification [5]. Related Malayalam work combines similarity measures with linguistic preprocessing and supervised classification [6]. A recent Kazakh study illustrates that method comparisons in agglutinative languages need labelled data and explicit evaluation protocols [7].

Table 1: Literature review

Study	Language	Relevant approach	Difference from present work
Gadag and Sagar [3]	Kannada	Trigram paraphrase generation	Generation and R-precision, not source-target screening
Prathibha and Padma [2]	Kannada	Rule-based lemmatization	No lemmatization is implemented here
Ebadulla et al. [4]	Kannada	Embeddings and transformers	No embedding model is implemented here
Kannada paraphrase study [5]	Kannada	Similarity features and classifiers	The present study has no labels or classifier
Biloshchytska et al. [7]	Kazakh	N-gram and semantic comparison	Uses purpose-built labelled duplicates

2.1 Plagiarism detection and paraphrasing

Plagiarism detection is complicated by paraphrasing because a reused passage may retain its central meaning while changing vocabulary, morphology, sentence order, syntax, or discourse structure. Barrón-Cedeño et al. developed the P4P corpus to analyze paraphrase mechanisms in plagiarism and reported that complex paraphrase phenomena reduce the effectiveness of automatic plagiarism detectors. Their analysis identified lexical substitutions and addition/deletion operations as frequent mechanisms in paraphrased plagiarism [11].

2.2 Character n-grams and stylometry

Character n-grams are language-independent textual features that can be extracted without extensive linguistic resources. In authorship-verification research, character n-gram frequency vectors have been used with cosine and min–max similarity measures to represent documents. Such representations are relevant to the present work because the implemented system also computes character-level bigram and trigram similarity using frequency-based cosine similarity [12].

2.3 Plagiarism in Indian languages

Deep-learning approaches have also been evaluated for paraphrase identification in Indian languages. Thenmozhi et al. used BERT, Universal Sentence Encoder, and Seq2Seq models for paraphrase classification on the DPIL corpus in Tamil, Malayalam, Hindi, and Punjabi. Their work is relevant as a semantic and supervised-learning direction for future development; however, these models and labelled datasets were not used in the present Kannada document-comparison system [13].

3 METHODOLOGY

3.1 Data

The system compares Kadyanata as source and Karavaliya Saviradondu Daivagalu as target. They are UTF-8 Kannada text files. The output report gives 1,388 source sentences and 219 target sentences after code-based segmentation.

Table 2: Input data details

Characteristic	Value
Source input	Kadyanata
Target input	Karavaliya Saviradondu Daivagalu
Source sentences	1,388
Target sentences	219

3.2 Implementation and preprocessing

The application is written in Python using tkinter, re, difflib.SequenceMatcher, collections.Counter, unicodedata, and math. It reads UTF-8 text. Sentence extraction applies NFKC normalization, collapses whitespace, splits on [।.!?]+, strips fragments, and removes empty fragments. N-gram extraction repeats NFKC normalization and whitespace collapse. Word n-grams are created by whitespace splitting. The code does not lowercase, remove punctuation or stop words, stem, lemmatize, perform POS tagging, retrieve sources, or apply semantic embeddings.

3.3 Similarity measures

For every target sentence, the code applies SequenceMatcher ratio against every source sentence and retains the highest score. A match is recorded when the best score is at least 0.60.

Sentence-match rate = matched target sentences / all target sentences × 100.

For character and word n-grams of n = 2 and n = 3, set-based Jaccard similarity is:

$$J(A,B) = |A \cap B| / |A \cup B|$$

Frequency-vector cosine similarity is :

$$\cos(x,y) = \Sigma_i x_i y_i / (\sqrt{\Sigma_i x_i^2}\ \sqrt{\Sigma_i y_i^2})$$

The program averages eight scores: Jaccard and cosine for character bigrams, character trigrams, word bigrams, and word trigrams.

Combined score = 0.40 × sentence-match rate + 0.60 × mean n-gram similarity. The program labels a 40% to < 70% score as moderate suspicion.

3.4 Workflow

The proposed workflow begins by loading the source and target Kannada text files in UTF-8 encoding. Both documents are then normalized using Unicode NFKC normalization, and repeated whitespace is reduced to a single space to improve consistency during comparison[8]. The normalized texts are segmented into sentences using the supported sentence-ending punctuation marks. For sentence-level analysis, each target sentence is compared with all source sentences, and the highest similarity score is retained. A target sentence is considered a potential match when its best similarity score is equal to or greater than the configured threshold of 0.60.

In parallel, the complete normalized source and target documents are processed to generate character-level and word-level bigrams and trigrams. For each n-gram representation, Jaccard similarity is calculated to measure the proportion of shared unique n-grams, while cosine similarity is calculated from n-gram frequency vectors to measure distributional overlap. The system produces eight n-gram similarity values: character bigram, character trigram, word bigram, and word trigram scores, each evaluated using both Jaccard and cosine measures. These eight values are averaged to obtain the overall n-gram similarity score.

Finally, the system combines the sentence-level similarity result and the average n-gram similarity using the implemented weighting scheme: 40% for the sentence-match rate and 60% for the average n-gram similarity. The resulting combined score is used to generate a plagiarism-screening status and a detailed report containing document statistics, similarity scores, matched sentences, and overlapping bigram and trigram phrases.

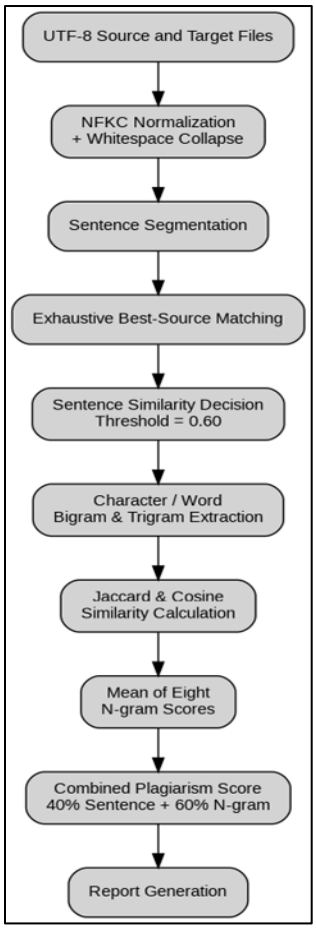

Figure 1: Workflow for Plagiarism Detection

4 EXPERIMENTAL SETUP

The system was implemented and executed using Python 3.11.9 on a Windows-based laptop. The application uses Python standard libraries, including Tkinter, `re`, `difflib`, `collections`, `unicodedata`, and `math`.

Table 3: Experimental setup

Parameter	Setting
Programming language	Python 3.11.9
Operating system	Windows
Hardware platform	Laptop
User interface	Tkinter
Libraries used	tkinter, re, difflib, collections, unicodedata, math
Analysis mode	Combined
Sentence similarity threshold	0.60
N-gram sizes	Bigram (n=2n=2n=2) and trigram (n=3n=3n=3)
Similarity measures	Jaccard similarity and cosine similarity
Combined-score weights	40% sentence similarity and 60% average n-gram similarity

5 RESULTS AND DISCUSSION

The report's calculations are internally consistent. Sentence-match rate = 138 / 219 × 100 = 63.0137%, rounded to 63.01%. The eight n-gram percentages sum to 288.64; dividing by 8 gives 36.08%. The combined score is 0.40 × 63.01 + 0.60 × 36.08 = 46.852%, rounded to 46.85%.

Table 4: Sentence level checking results

Outcome	Value (%)	Meaning
Sentence-match rate	63.01	138 of 219 target sentences had a best source match ≥0.60
Mean n-gram similarity	36.08	Arithmetic mean of eight whole-document n-gram scores
Combined screening score	46.85	Code-defined weighted score; moderate-suspicion status

Table 5: Bigram and Trigram checking results

Representation	Bigram Jaccard	Bigram cosine	Trigram Jaccard	Trigram cosine
Character	44.12	96.05	31.45	88.89
Word	4.28	15.43	2.17	6.25

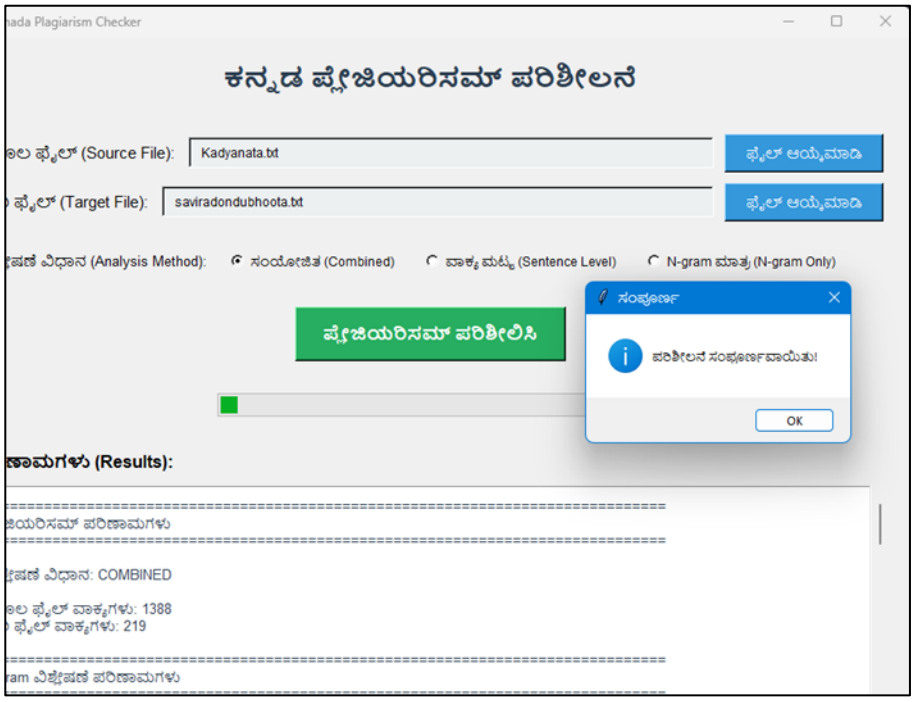

Figure 2: Output Screen

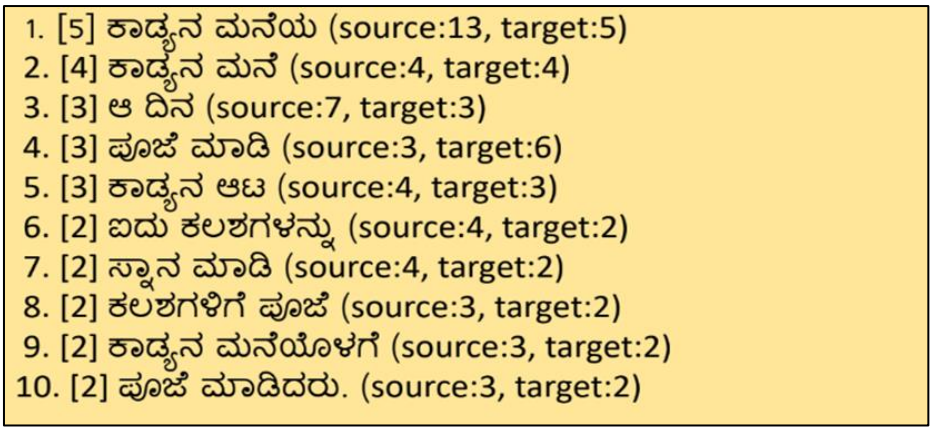

Figure 3: Top 10 matching phrases (Word Bigrams)

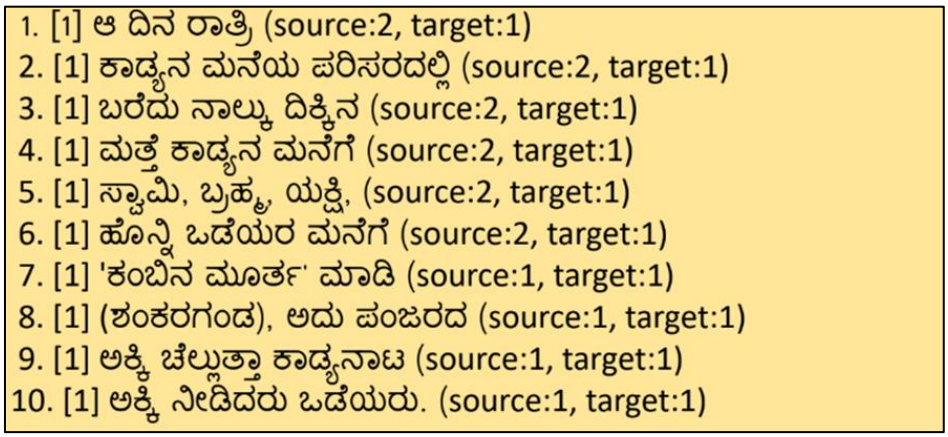

Figure 4: Top 10 matching phrases (Word Trigrams)

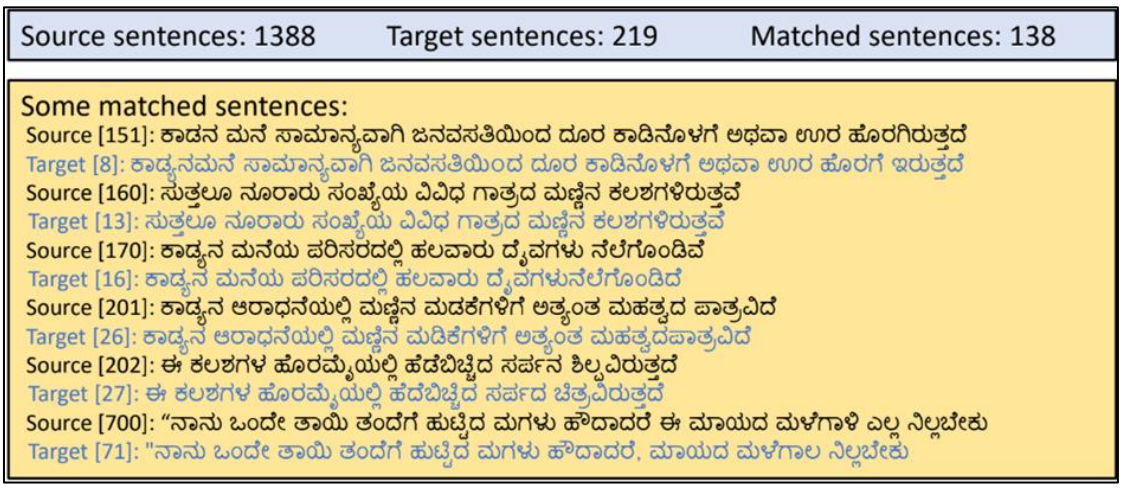

Figure 5: Sentence Level Checking

Figure 6 compares the eight n-gram similarity scores obtained for the source and target documents. Character-level n-gram measures show substantially higher similarity than word-level measures, particularly for cosine similarity. This figure specification is used because the numerical table is the supplied evidence.

Character-level cosine values were much higher than word-level scores in this run. This indicates greater frequency-distribution overlap for short character sequences than for exact word n-gram sequences. The most frequent shared

word bigram was "ಕಾಡ್ಯನ ಮನೆಯ" with overlap 5 (source 13; target 5) as shown in Figure 3. All top word trigrams had overlap count 1 as shown in Figure 4.

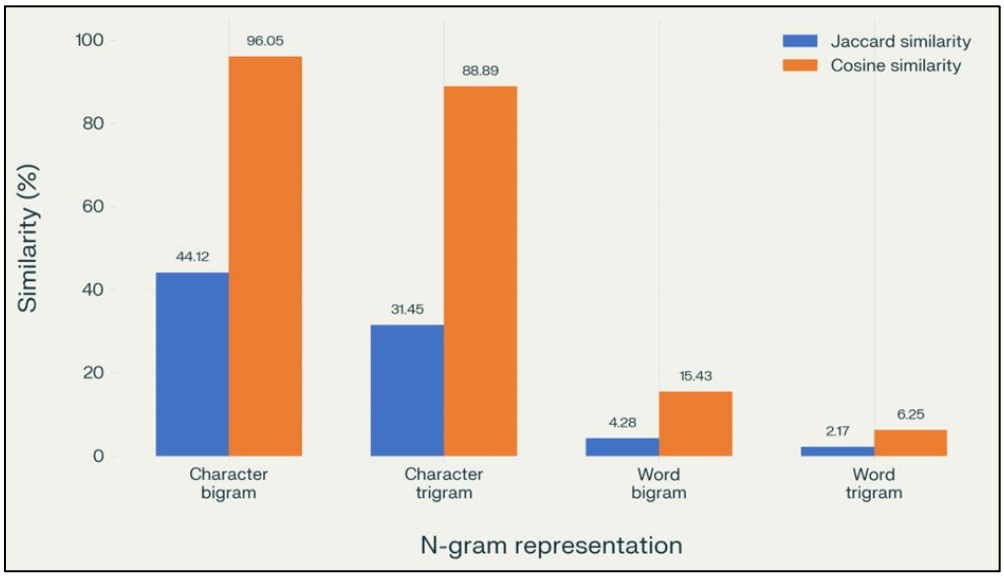

Figure 6: Comparison of character-level and word-level bigram and trigram similarity scores using Jaccard and cosine measures.

Figure 5 shows sentence matches from 60.00% to 100.00%. A sentence meeting the threshold is a candidate for human review, not proof of copying. Topic-specific vocabulary, repeated names, generic phrases, and shared context can increase lexical overlap. Conversely, heavily paraphrased passages may fall below a lexical threshold.

Future work

Future work should use a permission-cleared Kannada corpus with independently annotated exact reuse, edited reuse, paraphrase, and non-reuse examples. That would enable threshold calibration and held-out precision, recall, F1-score, and error analysis. Further experiments could compare raw-text processing with Kannada lemmatization, explore hybrid lexical-semantic models, add source retrieval, and test explanations that distinguish generic topical overlap from distinctive reuse.

6 CONCLUSION

The supplied code implements transparent Kannada similarity screening through best-match sentence comparison and character/word bigram-trigram Jaccard and cosine measures. In the documented run, the sentence-match rate was 63.01%, mean n-gram similarity was 36.08%, and the code-defined combined score was 46.85%. These values were reproduced from the report. The tool surfaces measurable lexical similarity indicators in the selected pair and provides evidence for manual review.

STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

Disclosure of conflict of interest

The authors declare no conflicts of interest regarding this manuscript.

REFERENCES

[1] Sajid M, Sanaullah M, Malik TS, Shuhidan SM, Fuzail M. Comparative analysis of text-based plagiarism detection techniques. PLoS One. 2025;20(4):e0319551. doi:10.1371/journal.pone.0319551.

[2] Prathibha RJ, Padma MC. Design of rule based lemmatizer for Kannada inflectional words. In: ICERECT; 2015. p. 264-269.

[3] Gadag AI, Sagar BM. N-gram based paraphrase generator from large text document. In: CSITSS; 2016. p. 91-94.

[4] Ebadulla DM, Raman R, Shetty HK, Mamatha HR. A comparative study on language models for the Kannada language. In: ICNLSP; 2021. p. 280-284.

[5] H M A, Mahesh J, Sairam K, Mamatha HR. Paraphrase detection in a low-resourced language: Kannada. 2023 IEEE 8th International Conference for Convergence in Technology (I2CT). 2023;doi:10.1109/12CT57861.2023.10126391

[6] Sindhu.L, Sumam Mary Idicula, Plagiarism detection in Malayalam language text using a composition of similarity measures.

[7] Biloshchytska S, Tleubayeva A, Kuchanskyi O, et al. Text similarity detection in agglutinative languages: A case study of Kazakh using hybrid N-gram and semantic models. Applied Sciences. 2025;15:6707. doi:10.3390/app15126707.

[8] Pawan Lahoti, Namita Mittal, and Girdhari Singh. 2022. A Survey on NLP Resources, Tools, and Techniques for Marathi Language Processing. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 22, 2, Article 47 (December 2022);https://doi.org/10.1145/3548457

[9] Muhammad Faraz Manzoor, Muhammad Shoaib Farooq,Adnan Abid. Stylometry-driven framework for Urdu intrinsic plagiarism detection: a comprehensive analysis using machine learning, deep learning, and large language models. Neural Computing and Applications (2025) 37:6479–6513, https://doi.org/10.1007/s00521-024-10966-w

[10] Alberto Barron-Cede, Marta Vila, M. Antonia Mart, Paolo Ross. Plagiarism Meets Paraphrasing: Insights for the Next Generation in Automatic Plagiarism Detection 2013. 2012 Association for Computational Linguistics. doi:10.1162/COLI a 00153

[11] Barrón-Cedeño A, Vila M, Martí MA, Rosso P. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. *Computational Linguistics*. 2013;39(4):917–947. doi:10.1162/COLI_a_00153

[12] Moshe Koppel, Yaron Winter. Determining If Two Documents Are Written by the Same Author. Journal of the Association for Information Science and Technology, 23 October 2013. DOI: 10.1002/asi.22954

[13] Thenmozhi D, Mahibha CJ, Kayalvizhi S, Rakesh M, Vivek Y, Poojesshwaran V. Paraphrase detection in Indian languages using deep learning. In: Anand Kumar M, et al., editors. *Proceedings of SPELLL 2022*. Communications in Computer and Information Science. Vol. 1802. Cham: Springer; 2023. p. 138–154. doi:10.1007/978-3-031-33231-9_9