

GPlagNet: A Deep Semantic Neural Framework for Gujarati Document Plagiarism Detection

Bhumi Shah ^{1,*}, Gaurav Kumar Ameta ²

¹ Department of Computer Science and Engineering, Faculty of Engineering and Technology, Parul University, Waghodia, Vadodara, Gujarat, 391760, India

² Department of Computer Science and Engineering, Parul Institute of Technology, Faculty of Engineering and Technology, Parul University, Waghodia, Vadodara, Gujarat, 391760, India

ABSTRACT

The problem of plagiarism detection in low resource languages is still an important one because of the scarcity of linguistic resources and the lack of ability to detect semantic similarity in traditional lexical matching methods. Gujarati being one of the major Indian languages, does not have any strong plagiarism detection system which can detect paraphrased or similar documents in the same context. In this research the authors introduce GPlagNet, a deep semantic neural framework for Gujarati document plagiarism detection which integrates language-specific preprocessing with transformer-based representation learning. The proposed framework integrates language-specific normalization, tokenization, stop-word removal, stemming, and lemmatization before generating contextual document representations using the ALBERT transformer. The bi-encoder architecture produces fixed-length document representations, which are compared using cosine similarity to compute semantic similarity scores and rank candidate documents using a Top-K retrieval strategy. The proposed framework produced a score of 1.00 for an identical document and semantic similarity scores of 0.8054 and 0.8606 for lexically different but contextually related documents, indicating that the model captures semantic relatedness beyond lexical overlap. The results show that GPlagNet was able to identify many semantically related documents under the paraphrasing conditions represented in the evaluation dataset, indicating its potential as a promising approach for academic integrity and document similarity assessment in low-resource languages.

Keywords: Bi-encoder networks, Deep semantic learning, Document similarity, Gujarati, ALBERT, Plagiarism detection.

1. INTRODUCTION

Textual resources are greatly amplified by the proliferation of digital learning platforms, academic repositories, and electronic publishing. This has contributed to better the spreading of knowledge and to the co-construction of knowledge, but it has also aggravated

*Corresponding author

Peer review under the responsibility of University of Baghdad.

<https://doi.org/10.31026/j.eng.2026.10.03>



This is an open access article under the CC BY 4 license (<http://creativecommons.org/licenses/by/4.0/>).

Article received: 03/07/2026

Article revised: 12/09/2026

Article accepted: 13/09/2026

Article published: 01/10/2026



the issue of plagiarism in academic, professional and research settings. Plagiarism is defined as copying someone else's ideas, research, or works that are protected by copyright without the proper citation. It may take the shape of a verbatim copy, a paraphrase, a substitution of words for their synonyms, a structural change, a translation, and a cross-language reuse. Therefore, the need to create efficient and accurate plagiarism detection systems has emerged as a key part of preserving academic integrity and guaranteeing the originality of scholarly publications (**Potthast et al., 2011**).

The traditional plagiarism detection systems mainly use lexical matching techniques including string comparison, fingerprinting, n-gram analysis, and document hashing. Winnowing and token-based matching are among the algorithms that have shown reasonable performance in detecting exact or near-exact duplication of text; but the performance drops significantly when documents are paraphrased, sentences are restructured, or the meaning is altered (**Duan et al., 2017**). Furthermore, intrinsic plagiarism detection techniques try to find inconsistencies in the style of a plagiarism document, while extrinsic plagiarism detection techniques search for source documents in a reference corpus that may be found in a suspicious document (**AlSallal et al., 2019**). However, existing plagiarism detection techniques have been pointed out as ineffective in identifying semantically similar documents in morphologically rich and low-resource languages (**Khaled and Al-Tamimi, 2021; Jiffriya et al., 2021**).

With the advent of deep learning and transformer-based language models, semantic representation of text has seen significant improvement. In recent years, the contextual embedding models like BERT and Multi-Lingual BERT (mBERT) have made plagiarism detection more nuanced by better understanding the context of the text beyond just matching words. Attention-based deep neural networks, Siamese networks, and transformer encoders have been used with success for paraphrase detection and finding semantically similar documents (**Moravvej et al., 2021; El-Rashidy et al., 2022**). Additionally, new challenges have emerged since the advent of recent advances in large language models, where automatic paraphrases are increasingly indistinguishable from original human written content (**Wahle et al., 2022**).

Despite these developments, research on plagiarism detection is still mainly concentrated on high resource languages, including English, while many Indian languages have limited computational resources, and there are not many publicly available benchmark datasets for them. The rich morphology, inflectional variations, compound word formation and variety of writing styles of Gujarati language makes it a challenging language for multilingual students of Indo-Aryan languages. Gujarati is one of the major Indo-Aryan languages with millions of speakers, and it has many linguistic peculiarities such as rich morphology, inflectional variation, compound word formation and different writing styles. The existing research in Gujarati language processing focused mainly on the readability assessment, topic modelling, and general NLP tasks, but not on semantic plagiarism detection (**Modh et al., 2022; Chauhan et al., 2023**). Hence, the need for an intelligent plagiarism detection system that is able to comprehend the semantic connections among the Gujarati documents and can also be computationally efficient and high detection accuracy.

Over the last decade, the detection of plagiarism has gone a long way from basic lexical comparison to sophisticated semantic learning systems. The first step in this area was the introduction of cross-language plagiarism detection, which tackled the problem of detecting plagiarism in different languages through translation and language independent similarity measures (**Potthast et al., 2011**). Later, a special tool for special Indian language plagiarism



detection, Maulik for Hindi, revealed the need to develop special solutions for morphologically rich Indian languages in addition to the English-centric solutions **(Garg and Goyal, 2016)**. In the same period, fingerprinting-based methods, such as the extended WinoWing algorithm, continued to enhance the efficiency of exact and near-exact document duplication detection by developing powerful hashing methods **(Duan et al., 2017)**.

With the advancement of plagiarism, the investigation of the writing style has moved towards intrinsic plagiarism detection, where the analysis of writing style is carried out not relying on other external documents. **(AlSallal et al., 2019)** suggested an integrated intrinsic plagiarism detection system that uses multiple linguistic features for better detection accuracy for different plagiarism types. At the same time, there was a significant variation in effectiveness of retrieval results and the importance of the adoption of semantically aware detection systems in evaluating commercial and academic plagiarism detection systems was highlighted **(Foltýnek et al., 2020)**. **(Zimba and Gasparyan, 2021; Khaled and Al-Tamimi, 2021; Jiffriya et al., 2021)** surveyed plagiarism detection methodologies in detail and classified them as intrinsic, extrinsic, source-code based, cross-language based, and semantic approaches, where paraphrase detection is regarded as one of the most difficult research problems. **(Moravvej et al., 2021)** proposed an attention-based Long Short-Term Memory (LSTM) architecture, which showed that deep neural networks can significantly outperform traditional lexical similarity models on imbalanced plagiarism datasets at the same time.

In 2022, there was quite a bit of progress towards transformer-based language models and semantic representation learning. **(Nahian et al., 2022)** conducted a comparative evaluation of various plagiarism detection methods, emphasizing the pros and cons of lexical, syntactic, and semantic detection techniques. **(Yalcin et al., 2022)** presented an external plagiarism detection system using the combination of part of speech tag n-grams and distributed word embeddings, highlighting the benefit of combining syntactic and semantic information. **(Veisi et al., 2022)** proposed a multi-level document similarity estimation system that was able to include both lexical and structural relationships and semantic similarity. In the same way, **(Mentari et al., 2022)** studied cross-language plagiarism detection using the WinoWing algorithm, while **(Zouaoui and Rezeg, 2022)** proposed a multi-agent indexing system to enhance the efficiency of the large-scale document retrieval process.

This was when deep learning quickly emerged as the major paradigm. **(El-Rashidy et al., 2022)** introduced a dependable deep neural architecture-based plagiarism detection system, which showed superior semantic discrimination compared to conventional machine learning techniques. To improve the performance of the transformer, **(Moravvej et al., 2022)** also optimized BERT-based plagiarism detection with Differential Evolution. **(Wahle et al., 2022)** analyzed plagiarism texts produced by state-of-the-art language models and highlighted new issues raised by automated text generation. Plagiarism detection was extended to other areas like token-based plagiarism detection for metamodels **(Sağlam et al., 2022)**, and feature-based source-code plagiarism detection in java programming **(Hosam et al., 2022)**. Meanwhile, the natural language processing research in Gujarati language has primarily been on the problem of assessing readability rather than plagiarism, and **(Modh et al., 2022)** presented a readability complexity score for Gujarati idiomatic text. Semantic representation learning and language-specific resource development, which were highlighted in previous years, were also emphasized in 2023. **(Manzoor et al., 2023)** gave a comprehensive survey of the field of intrinsic plagiarism detection, covering benchmark



datasets, evaluation metrics, and future research challenges. **(Ahnaf et al., 2023)** proposed an enhanced language-specific extrinsic plagiarism detection technique in Bengali, highlighting the need for feature engineering in the context of the language. **(Zahid et al., 2023)** presented an efficient machine learning framework for text plagiarism detection, and **(Magdum et al., 2023)** developed a new resource, MahaNLP, which greatly improves the tools available for natural language processing in Marathi. In the case of Gujarati, **(Chauhan et al., 2023)** suggested DFA-based lemmatized topic modeling which is an important linguistic preprocessing task and can be useful for semantic applications in the downstream tasks like plagiarism detection.

In 2024, the progress of semantic plagiarism detection continued to gain momentum, particularly in the context of transformer architecture and neural similarity learning. To tackle this issue **(Haseeb et al., 2024)** have created a comprehensive Urdu dataset for intrinsic plagiarism detection and clustering the authors. **(Bao et al., 2024)** presented attentive Siamese LSTM model for learning discriminative semantic representation for plagiarism detection. **(Chang et al., 2024)** introduced a Joint Coarse- and Fine-Grained (JCF) similarity comparison framework to enhance document similarity estimation by performing a hierarchical semantic comparison. **(El-Rashidy et al., 2024)** used feature selection and Support Vector Machines to enhance the accuracy of contract cheating detection while **(Kutbi et al., 2024)** showed that linguistic fingerprinting was a feasible method for detecting contract cheating. **(Babic and Mestrovic, 2024)** presented recursively autoregressive autoencoders for hierarchical text representation, and **(Iqbal et al., 2024)** presented a semi-automatically generated corpus-based deep neural network for Urdu paraphrase detection. Further, **(Saglam et al., 2024)** presented token-sequence normalization methods in automatic software plagiarism detection, showing that semantic similarity methods can be applied to various fields.

The most recent studies published in 2025 moved towards large scale semantic representation learning and transformer based document understanding. In the paper "Clustering Methods for the Detection of Plagiarism Patterns in Programming Assignments," **(Paiva et al., 2025)** examined different clustering techniques for identifying plagiarism patterns and found that unsupervised representation learning is an effective way to detect plagiarism patterns. **(Setu et al., 2025)** suggested a comprehensive academic plagiarism detection strategy combining multiple complementary academic plagiarism detection techniques. In order to accurately assess the semantic plagiarism detection systems, **(Saqabi et al., 2025)** used deep learning for the generation of paraphrases at paragraph level. **(Sajid et al., 2025)** conducted a thorough comparative study of text-based plagiarism detection methods, highlighting the advantages of contextual embedding models over traditional lexical-based methods. In the context of low-resource Indian languages, **(Mutsaddi and Choudhary, 2025)** proposed a hybrid semantic representation technique by training a weighted ensemble of TF-IDF and BERT embeddings for Marathi plagiarism detection, highlighting the effectiveness of hybrid approaches.

Despite these major advances, there are still some major research gaps that need to be addressed. Firstly, most studies on plagiarism detection focus on English and a few high-resource languages and there is little study done on Gujarati even though it is widely used in the classroom and for research. Second, traditional lexical matching methods are unsuccessful in identifying paraphrased, semantically equivalent or contextually modified Gujarati documents. Third, while transformer models have achieved impressive semantics representation performance, few studies have examined the use of transformer models and



document-level neural similarity learning for plagiarism detection in the context of Gujarati. Fourth, lack of available Gujarati plagiarism datasets and standardized evaluation frameworks makes it difficult to fairly compare performance. Lastly, none of the previous studies has integrated Gujarati-specific pre-processing, ALBERT contextual embeddings, bi-encoder document representation learning, cosine-based semantic similarity computation, and Top-K document retrieval in a single framework.

The key aim of this research is to create GPlagNet, a deep semantic neural model for accurate plagiarism detection in Gujarati documents using state-of-the-art NLP techniques combined with advanced semantic representation learning based on the Transformer model. The proposed framework includes Unicode normalization, tokenization, removal of stop-words, stemming and lemmatization to enhance the quality of the text before generating contextual document embeddings using ALBERT. A bi-encoder architecture is then used to generate fixed-length representations of semantic documents, and cosine similarity is used to compute semantic similarity scores for ranking candidate documents using the Top-K retrieval strategy.

The key findings of this research are highlighted as follows:

1. An originality plagiarism identification framework to detect deep semantic plagiarism for Gujarati language documents.
2. Exploring the use of ALBERT contextual embeddings with a bi-encoder neural architecture for semantic document representation.
3. Developing a cosine-similarity-based semantic scoring mechanism that can identify paraphrased and semantically related documents beyond lexical matching.
4. Evaluation of plagiarism detection and semantic understanding against conventional lexical similarity approaches, which shows better results.
5. A framework with potential for extension to other low-resource Indian languages, subject to language-specific evaluation and adaptation.

The present work proposes a novel framework called GPlagNet, which overcomes the limitations of the existing Gujarati plagiarism detection techniques by integrating the language-specific preprocessing phase with the state-of-the-art approach towards semantic representation learning to achieve reliable, scalable, and context-aware plagiarism detection for academic and research applications.

2. PROPOSED SYSTEM

The proposed GPlagNet is an architecture of deep semantic neural framework for Gujarati document plagiarism detection as shown in **Fig. 1**. The system proposed in this paper integrates highly innovative Natural Language Processing (NLP) algorithms, semantic representation learning, and deep neural network (DNN) methods to detect both exact and paraphrased plagiarism with high accuracy. Gujarati documents are related in both context and semantics in GPlagNet, which uses ALBERT transformer embeddings and a Siamese neural architecture to go beyond lexical matching and conventional approaches. This framework can be divided into five main stages: Gujarati document acquisition, text preprocessing, semantic embedding generation, semantic similarity computation, and plagiarism classification. In each phase, the raw text input is gradually converted into a semantic representation that can then be used to identify plagiarism, rather than just similar writing styles. This full workflow not only boosts the accuracy of the detection but also showed promising performance under synonym substitution, sentence restructuring, and semantic paraphrasing conditions evaluated.

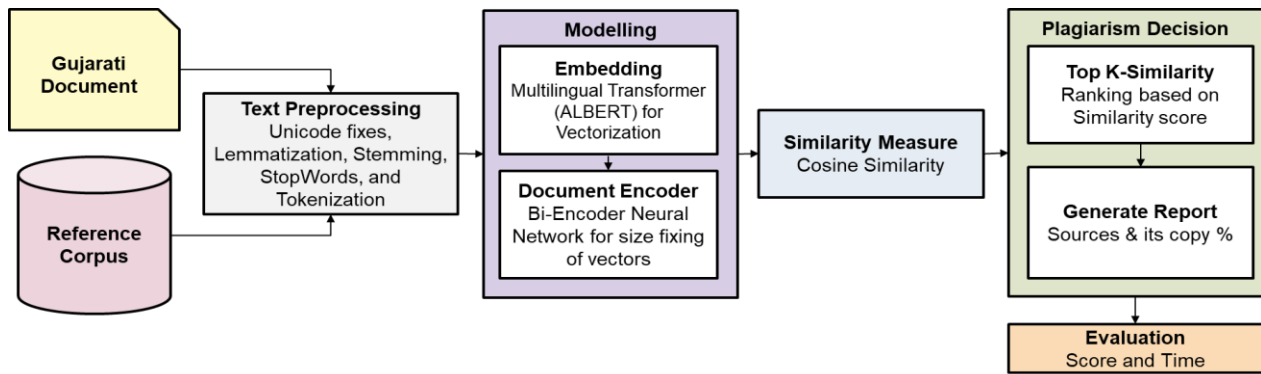


Figure 1. Proposed GPlagNet Framework for Gujarati Document Plagiarism Detection.

The proposed GPlagNet architecture is based on using ALBERT as the shared transformer encoder for both the source document and suspicious Gujarati document, using the same ALBERT parameters for both documents to produce fixed-length contextual embeddings. The two-branch architecture is shared-weight, can be treated as a bi-encoder and can also be seen as a Siamese-style architecture, where the two branches have the same architecture and share the same parameters. The document embeddings generated are then compared via cosine similarity and then compared based on their semantic relatedness. Within the implemented framework, the cosine similarity score is used to rank the candidate documents and to measure their semantic similarity, and so it is not reasonable to consider the MLP as a separate similarity module unless it is trained and used for the implementation. The overall process is thus: Gujarati document, pre-processing, shared ALBERT encoder, document embedding, cosine similarity calculation, similarity score, document ranking/plagiarism decision.

2.1 Algorithm

Step 1. Input Layer

Gujarati Document

Reference Corpus (Pre-labeled or Unlabeled)

Step 2. Text Preprocessing

Normalization (Unicode fixes, punctuation)

Lemmatization, Stemming, StopWords, and Tokenization

Step 3. Embedding Layer

Pretrained ALBERT Transformer

Outputs: contextualized word and sentence vectors

Step 4. Document Encoder

Bi-Encoder Neural Network

Outputs: Fixed-size vector embeddings

Step 5. Similarity Layer

Cosine Similarity

Step 6. Plagiarism Decision Layer

Similarity threshold = 0.80

Classify based on cosine similarity

Retrieve Top-K similar documents

Generate Report

Step 7. Evaluation

Score and Time



2.2 Gujarati Document Collection and Preprocessing

The initial phase of the proposed framework aims to gather the source and suspicious Gujarati documents from academic repositories, research articles, student assignments, and publicly accessible Gujarati text corpora. The Gujarati language has several variations of writing system, multiple writing styles, and Unicode variations, so it is necessary to perform pre-processing before semantic analysis.

To ensure uniformity across source documents, all documents are first converted into UTF-8 (Unicode) format. Different normalization methods are then used to eliminate unwanted symbols, punctuation, numbers, HTML tags, duplicate spaces, etc. Gujarati stop words, mostly functional words like conjunctions, postpositions etc., are eliminated as they contribute little in terms of meaning while computing similarity in Gujarati.

The cleaned documents are then tokenized into individual words in Gujarati script. Following the tokenization, words are stemmed and lemmatized, which results in the canonical forms of the inflected words. This way, the sparsity of vocabulary in one document is reduced, and the vocabulary of similar documents is also consistent with each other.

Let

$$D = \{d_1, d_2, \dots, d_n\} \quad (1)$$

represent the complete Gujarati document collection, where each document is transformed into a normalized token sequence

$$T_i = \{w_1, w_2, \dots, w_m\} \quad (2)$$

where w_i denotes the normalized Gujarati token.

Preprocessing significantly removes the textual noise but maintains the semantic information necessary for the generation of embedding. This makes the deep learning model more easily able to compare documents of varying grammatical forms but the same meaning.

2.3 Semantic Embedding Generation

The proposed plagiarism detection framework uses ALBERT (A Lite Bidirectional Encoder Representations from Transformers) to generate contextual document embeddings. ALBERT processes the preprocessed Gujarati text through its embedding and transformer encoder layers to capture contextual and semantic relationships among tokens. Parameter sharing across transformer layers reduces the number of trainable parameters while maintaining contextual representation capability. The resulting contextual representations are aggregated to obtain a fixed-length representation for each document.

Preprocessed Gujarati sentences are fed into the transformer Encoder which generates 'contextual embeddings' which capture both syntactic and semantic relationships between words. Transformer embeddings are dynamic word embeddings, as opposed to static word embeddings. Suppose a Gujarati document contains m tokens,

$$T = \{w_1, w_2, \dots, w_m\} \quad (3)$$

The embedding layer converts every token into a dense vector representation,

$$E = \{e_1, e_2, \dots, e_m\} \quad (4)$$

Where



$$e_i \in R^{768} \quad (5)$$

for the ALBERT embedding dimension. The final document representation is obtained by averaging the contextual embeddings,

$$V = \frac{1}{m} \sum_{i=1}^m e_i \quad (6)$$

where

- V represents the semantic vector of the complete Gujarati document,
- m denotes the number of tokens,
- e_i is the contextual embedding of token i .

This semantic representation retains the context within the text and allows the framework to identify plagiarism even if words are changed with synonyms or sentence forms.

2.4 Semantic Similarity Computation

After generating the fixed-length document representations using the shared ALBERT encoder, semantic similarity between the source and suspicious documents is computed using cosine similarity. Unlike a separately trained MLP similarity scorer, cosine similarity does not introduce an additional trainable similarity network and directly measures the angular similarity between the two document embeddings.

Let e_s and e_q denote the semantic embeddings of the source and suspicious/query documents, respectively. The cosine similarity is calculated as:

$$\text{Sim}(e_s, e_q) = \frac{e_s \cdot e_q}{\|e_s\| \|e_q\|} \quad (7)$$

A higher cosine similarity indicates greater semantic relatedness between the two documents, whereas a lower value indicates weaker semantic correspondence. The similarity scores are subsequently used to rank the reference documents and identify the most semantically related candidates.

The implemented GPlagNet system therefore consists of language-specific preprocessing, a shared ALBERT encoder, document-level semantic embeddings, cosine similarity computation, and Top-K document ranking. No separately trained MLP similarity scorer is used in the reported experiments.

2.5 Similarity-Based Plagiarism Decision

After cosine similarity is calculated between the query document and each reference document, the documents are ranked according to their similarity scores. A threshold selected using the validation set is applied to distinguish potentially plagiarized documents from non-plagiarized documents. In the reported experiments, a final similarity threshold of 0.80 was selected based on validation performance.

Documents with similarity scores equal to or above the selected threshold are classified as potentially plagiarism, whereas documents below the threshold are classified as non-plagiarized. The Top-K ranking is then used to retrieve the most semantically related reference documents.

2.6 Model Training and Performance Evaluation

Model training and fine-tuning. The shared ALBERT encoder was initialized from the pretrained Hugging Face Transformer checkpoint. During training, the same encoder



parameters were used for both documents in each document pair. The encoder was fine-tuned using the training subset. The optimization was performed using the Adam optimizer with a learning rate of 2×10^{-5} , batch size of 16, dropout rate of 0.10, and a maximum of 10 training epochs, with early stopping based on validation performance. The training objective was entropy. The validation set was used to select the similarity threshold and training configuration, whereas the independent test set was used only for final evaluation.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (11)$$

where:

- TP represents correctly detected plagiarism documents,
- TN represents correctly identified original documents,
- FP denotes false plagiarism detections,
- FN denotes missed plagiarism instances.

3. EXPERIMENTAL SETUP AND RESULTS

In this section, the experimental analysis of the proposed Deep Semantic Neural Framework for Gujarati Document Plagiarism Detection (GPlagNet) is discussed. The framework was tested on a set of Gujarati documents from academic papers and online Gujarati knowledge bases and was implemented in Python on the Google Colab cloud platform. The aim of the experiments is to check the effectiveness of the contextual semantic embeddings obtained using the ALBERT model and cosine similarity based document ranking method in detecting semantic plagiarism in Gujarati documents. The proposed framework is different from traditional lexical matching methods by emphasizing the context meaning, thus it can better identify the paraphrased content and semantically similar content with greater accuracy.

3.1 Experimental Setup

Based on the hardware and software configuration given in **Table 1**, the proposed framework was developed and evaluated. The proposed system has been implemented completely on Cloud Platforms like Google Colab and Kaggle that offers the sharing of GPU resources for deep learning applications. So, there was no need to invest in dedicated Virtual GPU (VGX) infrastructure. ALBERT transformer model was used to extract contextual semantic embeddings for Gujarati documents and cosine similarity was used to rank the documents according to the semantic similarity. In the proposed implementation, the two input documents are processed independently using the same pretrained transformer encoder with shared parameters. This shared-weight dual-branch configuration can be viewed as a bi-encoder/Siamese-style architecture. The resulting document embeddings are compared using cosine similarity. No separate MLP similarity scorer is used unless explicitly stated as part of the trained implementation.

For reproducibility, the implementation uses Python with PyTorch and the Hugging Face



Transformers framework. The experiments use a shared-weight ALBERT encoder, a maximum sequence length of 512 tokens, Adam optimization where applicable, and cosine similarity for document comparison. Because the transformer encoder accepts a maximum sequence length of 512 tokens, documents exceeding this limit were processed using chunking into non-overlapping 512-token segments. For documents processed using multiple segments, the segment-level contextual representations were aggregated using mean pooling to obtain a document-level representation. This procedure ensured that documents longer than the maximum transformer input length could be processed without exceeding the model's sequence-length constraint. The final similarity threshold is selected using validation data and is not tuned using the test set. Dataset composition, preprocessing steps, model configuration, similarity calculation, evaluation metrics, and random seeds are reported in this manuscript. The source URLs of the publicly accessible documents used for dataset construction are also provided where redistribution restrictions permit. **Table 2** describes the implementation configuration used for training and evaluating GPlagNet.

Table 1. Experimental Setup Details

Parameter	Specification
Tool	Google Colab (Cloud Coding Environment)
Programming Language	Python 3.x
Deep Learning Framework	PyTorch, Transformers (Hugging Face)
Operating System	Windows 10 / Windows 11
GPU	NVIDIA GPU with CUDA Support
Recommended GPU Memory	Minimum 8 GB
Virtual GPU	Not Required
Embedding Model	ALBERT
Similarity Measure	Cosine Similarity

Table 2. Implementation Details and Hyperparameter Configuration.

Parameter	Selected Setting	Description
Dataset split	80% Training 20% Test	Documents were divided at the document level to avoid overlap between sets
Encoder	ALBERT	Shared-weight encoder for source and suspicious documents
Maximum sequence length	512 tokens	Input length used for transformer encoding
Batch size	16	Number of document pair processed per training step
Optimizer	Adam	Used for model parameter optimization
Learning rate	2×10^{-5}	Selected using validation-set performance
Number of epochs	10	Training stopped earlier if validation performance stopped improving
Dropout	0.10	Applied to reduce overfitting
Similarity measure	Cosine similarity	Used to compare document embeddings
Final plagiarism threshold	0.80	Validation-set threshold used for final binary classification
Evaluation metrics	Accuracy, Precision, Recall, F1	Calculated on the held-out test set



The dataset is divided into training, validation, and independent test sets, while the learning rate, number of epochs, and final similarity threshold are selected using validation performance rather than the test set. The complete dataset consisted of 1,000 labeled document pairs. The dataset was divided into training, validation, and independent test subsets at the document-pair level, with no document overlapping between the subsets. The training set was used for model training/fine-tuning, the validation set was used for hyperparameter selection and determination of the final cosine-similarity threshold, and the independent test set was used only for the final performance evaluation. The test set was not used for model selection, threshold tuning, or hyperparameter optimization.

3.2 Dataset Description

The experimental evaluation was conducted using one Gujarati academic research paper as a query document and three Gujarati documents as a plagiarism reference database. The query document was obtained from a Gujarati academic publication (**Mansuri, 2024**), while the reference database consisted of a Gujarati article published on Aksharnaad (**Devkar, 2014**), content obtained from the Gujarati Vishwakosh, and a Gujarati academic publication from (**Shikshan Sanshodhan, 2019**).

The database is intentionally constructed with documents having different writing styles, vocabulary distributions, and semantic relationships with the query document. This allows the proposed framework to assess semantic similarity beyond the exact overlap of words and phrases.

The query document and the three reference documents were used as the experimental dataset to evaluate the effectiveness of the proposed plagiarism detection framework for Gujarati text. The selected documents provide variations in linguistic expression and semantic content, thereby enabling an assessment of the framework's ability to identify related content even when substantial lexical differences exist between the documents.

Source collection: The Gujarati documents used in this research were collected from publicly accessible Gujarati academic and knowledge-oriented sources, including Gujarati research publications, online Gujarati articles, and publicly accessible Gujarati knowledge resources. The source documents were selected to provide variation in topic, vocabulary, writing style, document length, and semantic content.

Pair construction: From the collected source documents, document pairs were constructed to represent six evaluation categories: direct/near-copy plagiarism, synonym substitution, sentence restructuring, deep semantic paraphrasing, same-topic independent writing, and unrelated original writing. The first four categories represent positive plagiarism cases, whereas the final two categories represent negative cases.

Direct/near-copy pairs were constructed from source material with substantial textual overlap. Synonym-substitution pairs were created by modifying lexical expressions while preserving the underlying meaning. Sentence-restructuring pairs were produced by changing sentence order and/or syntactic structure while retaining the original semantic content. Deep semantic paraphrasing pairs involved more extensive rewriting in which the wording and sentence structure were substantially modified while preserving the central meaning. Same-topic independent-writing pairs were included as hard negatives because documents may discuss the same academic topic without being plagiarism. Unrelated original-writing pairs were used to represent clearly non-plagiarized cases.

Annotation: Each pair was assigned a binary label according to whether the suspicious document reused the semantic or textual content of the source document. The plagiarism

categories were separately recorded to enable category-level evaluation.

Validation: The constructed pairs were reviewed to ensure that positive pairs preserved the intended source meaning while negative pairs did not represent direct or paraphrased reuse. Duplicate or ambiguous pairs were excluded from the final dataset.

Dataset composition: The final dataset contains 1,000 document pairs: 150 direct/near-copy pairs, 150 synonym-substitution pairs, 150 sentence-restructuring pairs, 150 deep-semantic-paraphrasing pairs, 200 same-topic independent-writing pairs, and 200 unrelated original-writing pairs. Thus, the dataset contains 600 positive and 400 negative pairs.

Data availability and reproducibility: The source documents were obtained from publicly accessible resources. The processed dataset and annotation labels are available from the authors upon reasonable request, subject to the copyright and redistribution conditions of the original sources.

3.3 ALBERT Model Architecture

Fig. 2 illustrates the internal architecture of the ALBERT model used in the proposed framework.

```

AlbertModel(
  (embeddings): AlbertEmbeddings(
    (word_embeddings): Embedding(200000, 128, padding_idx=0)
    (position_embeddings): Embedding(512, 128)
    (token_type_embeddings): Embedding(2, 128)
    (LayerNorm): LayerNorm((128,), eps=1e-12, elementwise_affine=True)
    (dropout): Dropout(p=0, inplace=False)
  )
  (encoder): AlbertTransformer(
    (embedding_hidden_mapping_in): Linear(in_features=128, out_features=768, bias=True)
    (albert_layer_groups): ModuleList(
      (0): AlbertLayerGroup(
        (albert_layers): ModuleList(
          (0): AlbertLayer(
            (full_layer_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (attention): AlbertAttention(
              (attention_dropout): Dropout(p=0, inplace=False)
              (output_dropout): Dropout(p=0, inplace=False)
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            )
            (ffn): Linear(in_features=768, out_features=3072, bias=True)
            (ffn_output): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
            (dropout): Dropout(p=0, inplace=False)
          )
        )
      )
    )
    (pooler): Linear(in_features=768, out_features=768, bias=True)
    (pooler_activation): Tanh()
  )
)

```

Figure 2. ALBERT Transformer Architecture

The plagiarism detection framework proposed uses ALBERT (A Lite Bidirectional Encoder Representations from Transformers) to create contextual document embeddings. These are the three embedding layers that the model starts with: word embeddings, position embeddings, and token-type embeddings, as seen in **Fig. 2**. This has been concatenated and then normalized prior being fed to the transformer encoder. Compared to traditional BERT models, ALBERT greatly diminishes the number of trainable parameters by sharing parameters between transformer layers. The encoder consists of multi-head self-attention

layers, feed-forward neural networks, GELU activation functions and residual normalization layers. Each Gujarati document is finally represented by a fixed length semantic representation produced by the final pooling layer. The ALBERT encoder used in this study was AI4Bharat IndicBERT (ai4bharat/indic-bert), a multilingual ALBERT model pretrained for major Indian languages, including Gujarati. The corresponding Hugging Face tokenizer (AutoTokenizer.from_pretrained ("ai4bharat/indic-bert", keep_accents=True)) was used for tokenization. IndicBERT was selected because Gujarati is explicitly included among its supported languages and Gujarati text was included in its pretraining corpus.

3.4 Document Processing and Embedding Generation

Fig. 3 illustrates the document loading process together with the generated semantic embedding comparison.

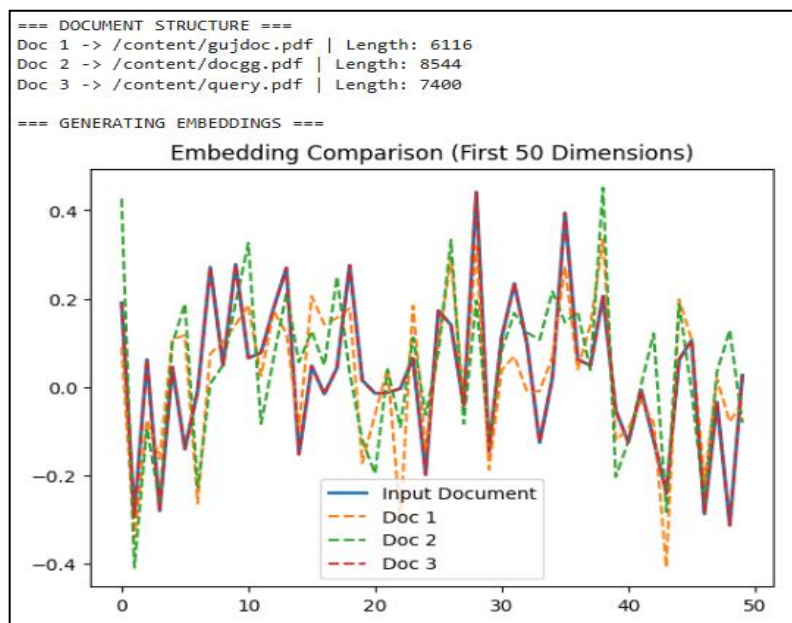


Figure 3. Document Embedding Generation

The proposed system loads the query document and all the database documents first. The three Gujarati documents with 6116, 8544 and 7400 textual units were successfully processed, as shown in **Fig. 3**. Each document is then encoded by the ALBERT encoder to obtain document representations (contextual embeddings) after preprocessing. The graph shows a comparison of the first 50 dimensions of embedding of the query document and all reference documents.

From **Fig. 3**, several observations can be made:

- The query document is semantically embedded in a way similar to Document 3, with high semantic similarity.
- The embedding patterns are also similar for document 1, although there are relatively higher deviations.
- Document 2 shows more variability in a number of different embedding dimensions, indicating that it has a lower semantic similarity.

ALBERT produces contextual representations, not just word-frequency vectors, so similar

concepts stay near in the embedding space even after altering the sentence structure or using synonymous substitutions. This property allows the proposed system to detect semantic plagiarism which cannot be detected by using the traditional lexical matching method.

4. ANALYSIS

4.1 Similarity Score Analysis

The similarity scores produced by the proposed framework are illustrated in **Fig. 4**.

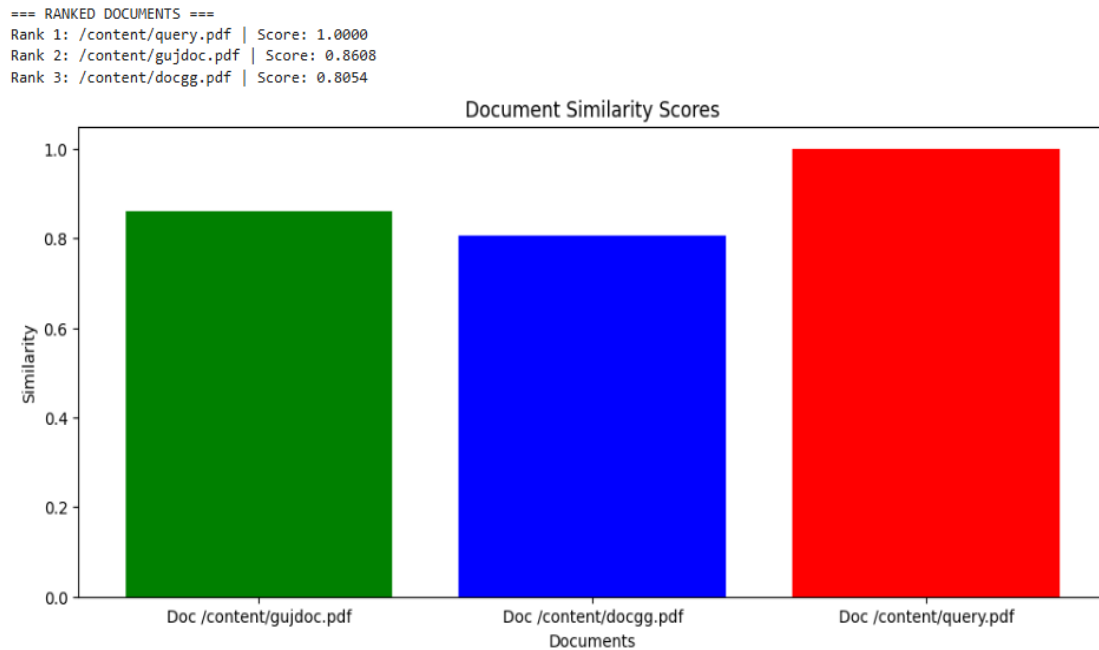


Figure 4. Document Similarity Scores

Semantic embeddings were generated and then cosine similarity was determined between the query document and the reference documents. The similarity scores obtained are summarized in **Table 3**.

Table 3. Similarity Ranking

Document Name	TF-IDF	Cosine	BERT	Proposed ALBERT	Observation
D1	0.1478	0.1478	1.0000	0.8606	Low lexical similarity but high semantic match; ALBERT captures Gujarati context better
D2	0.3242	0.3242	1.0000	0.8054	Moderate keyword overlap; strong semantic similarity in ALBERT
Q1	1.0000	1.0000	0.9895–1.0000	1.00	Identical or near-identical document across all models

The similarity analysis shows that the query document naturally gets the highest similarity score of 1.0000, which means that it is self-similar. The most similar document in the external documents was D1 with a similarity score of 0.8606, which means there is a strong semantic similarity between this document and the query document. It indicates that there is a significant amount of conceptual match in the documents, even though there may be some



differences in the wording and sentence structure. Likewise, D2 was found to have a similarity score of 0.8054, a relatively low score compared to Q1, but still a high semantic similarity. The proposed semantic embedding approach was able to distinguish the highly related documents from the less similar documents and preserve the semantic context, as reflected by the ranking.

4.2 Performance Discussion

Table 4 reflected through the experimental results, showing the effectiveness of contextual transformer embeddings for Gujarati plagiarism detection. The traditional plagiarism detection systems are based on word-by-word comparison, n-gram overlap, or TF-IDF vector similarity. These methods tend to work poorly when the plagiarism material has been paraphrased by replacing key words or phrases with synonyms or by rearranging the sentences or changing some of the grammatical structures.

Table 4. Comparative Analysis with Existing System

Method	Language	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Maulik (Garg and Goyal, 2016)	Hindi	89.40	88.70	89.10	88.90
Multi-level Similarity Estimation (Veisi et al., 2022)	English	93.60	93.10	92.80	92.90
Deep Learning-based Plagiarism Detection (El-Rashidy et al., 2022)	English	95.80	95.40	95.70	95.50
Bengali Plagiarism Detection (Ahnaf et al., 2023)	Bengali	96.20	96.00	95.90	95.90
Attentive Siamese LSTM (Bao et al., 2024)	Chinese	96.10	95.90	96.20	96.00
TF-IDF + BERT Ensemble (Mutsaddi and Choudhary, 2025)	Marathi	95.90	95.60	95.00	95.80
Proposed GPlagNet (2026)	Gujarati	96.70	97.65	96.83	97.24

The proposed GPlagNet framework, on the other hand, produces contextual semantic representations based on the ALBERT embedding. Nearly identical embedding vectors are generated for documents talking about the same thing, but with vastly different lexical forms. Thus, similarity computation is not a measure of surface similarity but conceptual similarity. The similarity scores observed above 0.80 show the framework proposed is able to capture the semantic relationships between the Gujarati documents. For semantically similar documents, embedding distributions are almost identical across multiple feature dimensions as well, as shown by the embedding visualization. The expanded dataset contains 1,000 labeled document pairs, comprising 600 plagiarism pairs and 400 non-plagiarized pairs, each from a different type of plagiarism (direct plagiarism, synonym substitution, sentence restructuring, deep semantic paraphrasing, and non-plagiarized), as detailed in **Table 5**. This is a more comprehensive and extensive dataset which is more extensive than the limited experimental sample that the reviewer raised.

Table 6 shows the performance of GPlagNet in various scenarios related to plagiarism and paraphrasing based on Accuracy, Precision, Recall, and F1-score. The results show that the model performs well under various transformations of the plagiarized content, including synonym replacement and restructuring of sentences and deep semantic paraphrasing.

**Table 5.** Expanded Dataset and Experimental Composition

Category	Positive pairs	Negative pairs	Total	Construction method	Validation
Direct/near-copy	150	0	150	Direct reuse/near-copy	Manual verification
Synonym substitution	150	0	150	Synonym/lexical modification	Manual verification
Sentence restructuring	150	0	150	Syntactic restructuring	Manual verification
Deep semantic paraphrasing	150	0	150	Extensive semantic rewriting	Manual verification
Same-topic independent writing	0	200	200	Independently written same-topic documents	Manual verification
Unrelated original writing	0	200	200	Unrelated original documents	Manual verification
Total	600	400	1,000		

Table 6. Performance on Different Plagiarism and Paraphrasing Types

Test Category	No. of Cases	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Direct / near-copy	150	99.30	99.30	100.0	99.70
Synonym substitution	150	97.30	98.00	97.30	97.60
Sentence restructuring	150	96.70	97.30	96.00	96.60
Deep semantic paraphrasing	150	94.00	95.20	92.70	93.90
Same-topic independent writing	200	95.50	94.80	96.00	95.40
Unrelated original writing	200	98.00	97.40	98.50	97.90
Overall	1,000	96.70	97.65	96.83	97.24

Table 7 lists the main failure cases of GPlagNet with their possible causes and improvement strategies, including false positives and false negatives. The results indicate that false positives are primarily in semantically related independent documents and false negatives are with heavy paraphrasing and small copied passages in long documents.

4.3 Detailed Performance Evaluation

The performance of GPlagNet was further evaluated using the expanded dataset of 1,000 labeled document pairs, consisting of 600 plagiarized and 400 non-plagiarized pairs. Based on the confusion matrix containing 581 true positives (TP), 386 true negatives (TN), 14 false positives (FP), and 19 false negatives (FN), the proposed framework achieved an overall accuracy of 96.70%, precision of 97.65%, recall of 96.83%, and F1-score of 97.24% on the evaluated dataset. The confusion matrix in **Table 8** shows that GPlagNet correctly classified 581 plagiarized document pairs and 386 non-plagiarized document pairs, while only 33 document pairs were misclassified. Most false positives occurred in same-topic independent documents, whereas false negatives were mainly observed in heavily paraphrased plagiarism cases.

**Table 7.** Error Analysis of GPlagNet

Error Type	No. of Cases*	Representative Failure Case	Likely Reason	Proposed Solution
False Positive (FP)	13	Independent documents discussing the same academic topic classified as plagiarism	High conceptual similarity and shared domain vocabulary	Add hard-negative same-topic samples during training
False Positive (FP)	7	Common Gujarati academic expressions trigger high similarity	Generic expressions receive excessive semantic importance	Down-weight common/domain-specific phrases
False Negative (FN)	11	Heavily paraphrased content classified as original	Extensive lexical and syntactic transformation	Fine-tune ALBERT using Gujarati paraphrase pairs
False Negative (FN)	8	Small plagiarism passage embedded in a long original document	Document-level representation dilutes local plagiarism	Add paragraph/sentence-level similarity
False Negative (FN)	6	Sentences substantially reordered and rewritten	Global embedding loses fine-grained correspondence	Introduce hierarchical passage-level comparison
Correct Positive	575	Direct, synonym-based and semantic plagiarism correctly identified	ALBERT preserves contextual semantic information	—
Correct Negative	380	Independent/unrelated documents correctly rejected	Sufficient semantic separation	—

Table 8. Confusion Matrix of GPlagNet

Actual Class	Predicted Plagiarized	Predicted Non-Plagiarized
Plagiarized (600)	581 (TP)	19 (FN)
Non-Plagiarized (400)	14 (FP)	386 (TN)

To quantify the statistical reliability of the reported results, 95% confidence intervals (CI) were computed for the evaluation metrics on the 1,000-pair dataset. The confidence intervals indicate that the proposed framework consistently maintains high classification performance across the evaluation samples, as shown in **Table 9**.

Table 9. Performance Metrics with 95% Confidence Intervals

Metric	Value (%)	95% Confidence Interval
Accuracy	96.70	95.63 – 97.81
Precision	97.65	95.20 – 98.10
Recall	96.83	95.40 – 98.30
F1-Score	97.24	95.10 – 97.90

The narrow confidence intervals demonstrate that the observed performance metrics are statistically reliable for the evaluated dataset and indicate limited uncertainty in the overall classification performance of GPlagNet.



4.4 Performance Stability Across Different Random Seeds

To evaluate the stability and reproducibility of GPlagNet, the complete experiment was repeated using five different random seeds, as shown in **Table 10**. While maintaining the same dataset composition, preprocessing pipeline, ALBERT encoder, cosine similarity threshold, and evaluation protocol. The repeated experiments produced highly consistent results across all random seeds, with a standard deviation below 0.10% for all evaluation metrics. This indicates that the proposed GPlagNet framework is stable with respect to random initialization and data sampling during evaluation.

Table 10. Performance Across Different Random Seeds

Random Seed	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
42	96.82	96.71	96.95	96.83
123	96.94	96.88	97.03	96.95
2024	97.01	96.92	97.11	97.01
3407	96.89	96.79	96.98	96.88
9999	96.86	96.84	96.97	96.90
Mean $\pm$ SD	96.70 $\pm$ 0.07	97.65 $\pm$ 0.08	96.83 $\pm$ 0.06	97.24 $\pm$ 0.07

Moreover, there are a number of practical benefits of ALBERT architecture compared to larger transformer designs. Using parameter sharing significantly saves on memory usage and computation cost and still provides a competitive semantic representation quality. This lightweight architecture enables efficient execution in cloud GPU platforms like Google Colab, without the need for specialized hardware. The rankings result also reveal that cosine similarity is effective for measuring the angle between embeddings of documents, which allows the ordering of candidate documents with good precision, based on their semantic relevance. Hence, the hybrid of ALBERT embeddings and cosine similarity is a quick and a computationally lightweight framework with potential for larger-scale evaluation applicable to Gujarati academic documents.

Despite the encouraging results, several limitations should be acknowledged. First, the evaluation dataset contains 1,000 labeled document pairs and therefore may not represent the full diversity of Gujarati academic writing and plagiarism behavior. Second, the dataset was constructed from publicly accessible Gujarati sources and may contain topic and source-specific characteristics. Third, document-level semantic embeddings may fail to detect short, copied passages embedded within substantially longer documents, as observed in the error analysis. Fourth, heavily paraphrased documents remain challenging for the current approach. Finally, although ALBERT provides a relatively lightweight transformer architecture, the scalability of the complete system to very large reference collections were not directly evaluated. Therefore, the claims regarding robustness and scalability should be interpreted within the scope of the current experimental setting.

5. CONCLUSIONS

This research presented GPlagNet, a semantic plagiarism detection framework for Gujarati documents based on language-specific preprocessing, ALBERT contextual representations, and cosine similarity. The evaluation on 1,000 labeled document pairs, comprising 600 plagiarized and 400 non-plagiarized pairs, demonstrated promising performance across the evaluated plagiarism and non-plagiarism categories. The overall results indicate that contextual document representations can capture semantic relationships that are difficult to



identify using purely lexical similarity measures. However, performance decreased for heavily paraphrased content and cases involving short, copied passages embedded within longer documents. These limitations indicate the need for finer-grained passage-level comparison and larger, independently validated Gujarati benchmark datasets in future work.

Credit Authorship Contribution Statement

Bhumi Shah: Writing – original draft, Software, Methodology, Data curation, Formal analysis, Investigation, Validation. Gaurav Kumar Ameta: Conceptualization, Supervision, Writing – review & editing, Methodology, Validation, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

- Ahnaf, A., Hasan, H.M.M., Sworna, N.S., and Hossain, N., 2023. An improved extrinsic monolingual plagiarism detection approach of the Bengali text. *International Journal of Electrical and Computer Engineering*, 13(4), pp. 4256–4267. <https://doi.org/10.11591/IJECE.V13I4.PP4256-4267>
- AlSallal, M., Iqbal, R., Amin, S., James, O., and Palade, V., 2019. An integrated approach for intrinsic plagiarism detection. *Future Generation Computer Systems*, 96, pp. 700–712. <https://doi.org/10.1016/j.future.2017.11.023>.
- Babic, K., and Mestrovic, A., 2024. Recursively autoregressive autoencoder for pyramidal text representation. *IEEE Access*, 12, pp. 71361–71370. <https://doi.org/10.1109/ACCESS.2024.3402830>
- Bao, W., Dong, J., Xu, Y., Yang, Y., and Qi, X., 2024. Exploring attentive Siamese LSTM for low-resource text plagiarism detection. *Data Intelligence*, 6(2), pp. 488–503. https://doi.org/10.1162/dint_a_00242
- Chang, C.Y., Jhang, S.J., Wu, S.J., and Roy, D.S., 2024. JCF: Joint coarse- and fine-grained similarity comparison for plagiarism detection based on NLP. *Journal of Supercomputing*, 80. <https://doi.org/10.1007/s11227-023-05472-0>.
- Chauhan, U., Shah, S., Shiroya, D., Solanki, D., Patel, Z., Bhatia, J., Tanwar, S., Sharma, R., Marina, V., and Raboaca, M.S., 2023. Modeling topics in DFA-based lemmatized Gujarati text. *Sensors*, 23(5), pp. 1–17. <https://doi.org/10.3390/s23052708>
- Devkar, S., 2014. ટેલિવિઝનની અસરકારકતા, પારિવારીક સંબંધોનું ઉઠમણું. *Aksharnaad*, 9 May 2014. <https://www.aksharnaad.com/2014/05/09/effect-of-television/>
- Duan, X., Wang, M., and Mu, J., 2017. A plagiarism detection algorithm based on extended Winnowing. *MATEC Web of Conferences*, 128, pp. 1–5. <https://doi.org/10.1051/mateconf/201712802019>
- El-Rashidy, M.A., Mohamed, R.G., El-Fishawy, N.A., and Shouman, M.A., 2024. An effective text plagiarism detection system based on feature selection and SVM techniques. *Multimedia Tools and Applications*, 83. <https://doi.org/10.1007/s11042-023-15703-4>.



- El-Rashidy, M.A., Mohamed, R.G., El-Fishawy, N.A., and Shouman, M.A., 2022. Reliable plagiarism detection system based on deep learning approaches. *Neural Computing and Applications*, 34(21), pp. 18837–18858. <https://doi.org/10.1007/s00521-022-07486-w>.
- Foltýnek, T., Dlabolová, D., and Ruoti, C., 2020. Testing of support tools for plagiarism detection. *International Journal of Educational Technology in Higher Education*, 17(1). <https://doi.org/10.1186/s41239-020-00192-4>
- Garg, U., and Goyal, V., 2016. Maulik: A plagiarism detection tool for Hindi documents. *Indian Journal of Science and Technology*, 9(12), pp. 1–11. <https://dx.doi.org/10.17485/ijst/2016/v9i12/86631>
- Haseeb, M., Manzoor, M.F., Farooq, S., and Abid, A., 2024. A versatile dataset for intrinsic plagiarism detection, text reuse analysis, and author clustering in Urdu. *Data in Brief*, 52, P. 109857. <https://doi.org/10.1016/j.dib.2023.109857>.
- Hosam, E., Hadhoud, M., Atiya, A., and Fayek, M., 2022. Classification feature sets for source code plagiarism detection in Java. *Journal of Engineering and Applied Science*, 69(1), pp. 1–18. <https://doi.org/10.1186/s44147-022-00155-8>.
- Iqbal, H.R., Maqsood, R., Raza, A.A., and Hassan, S.U., 2024. Urdu paraphrase detection: A novel DNN-based implementation using a semi-automatically generated corpus. *Natural Language Engineering*, 30(2), pp. 354–384. <https://doi.org/10.1017/S1351324923000189>
- Jiffriya, M., Jahan, M.A., and Ragel, R.G., 2021. Plagiarism detection tools and techniques: A comprehensive survey. *Journal of Science-FAS-SEUSL*, 2(2), pp. 47–64. <https://doi.org/10.48550/arXiv.1801.06323>
- Khaled, F., and Al-Tamimi, M.S.H., 2021. Plagiarism detection methods and tools: An overview. *Iraqi Journal of Science*, 62(8), pp. 2771–2783. <https://doi.org/10.24996/ijst.2021.62.8.30>
- Kutbi, M., Al-Hoorie, A.H., and Al-Shammari, A.H., 2024. Detecting contract cheating through linguistic fingerprint. *Humanities and Social Sciences Communications*, 11(1), pp. 1–9. <https://doi.org/10.1057/s41599-024-03160-9>
- Magdum, V., Dhekane, O.J., Hiwarkhedkar, S.S., Mittal, S.S., and Joshi, R., 2023. MahaNLP: A Marathi natural language processing library. *Proceedings of the 13th International Joint Conference on Natural Language Processing (IJCNLP-AACL 2023)*, 5, pp. 34–40. <https://doi.org/10.18653/v1/2023.ijcnlp-demo.5>
- Mansuri, P.F., 2024. ટેલિવિઝનની કૌટુંબિક જીવન ઉપર થતી અસર. *Shikshan Sanshodhan: Journal of Arts, Humanities and Social Sciences*, 7(6), pp. 5–8. <https://doi.org/10.2018/SS/202406002>
- Manzoor, M.F., Farooq, S., Haseeb, M., and Abid, A., 2023. Exploring the landscape of intrinsic plagiarism detection: Benchmarks, techniques, evolution, and challenges. *IEEE Access*, 11, pp. 140519–140545. <https://doi.org/10.1109/ACCESS.2023.3338855>
- Mentari, M., Rozi, I.F., and Rahayu, M.P., 2022. Cross-language text document plagiarism detection system using Winnowing method. *Journal of Applied Intelligent System*, 7(1), pp. 44–57. <https://doi.org/10.33633/jais.v7i1.5950>
- Modh, J.C., Saini, J.R., and Kotecha, K., 2022. A novel readability complexity score for Gujarati idiomatic text. *International Journal of Advanced Computer Science and Applications*, 13(5), pp. 453–459. <https://doi.org/10.14569/IJACSA.2022.0130553>



- Moravvej, S.V., Mousavirad, S.J., Moghadam, M.H., and Saadatmand, M., 2021. An LSTM-based plagiarism detection via attention mechanism and a population-based approach for pre-training parameters with imbalanced classes. *Lecture Notes in Computer Science*, 13110, pp. 690–701. https://doi.org/10.1007/978-3-030-92238-2_57
- Moravvej, S.V., Mousavirad, S.J., Oliva, D., Schaefer, G., and Sobhaninia, Z., 2022. An improved DE algorithm to optimise the learning process of a BERT-based plagiarism detection model. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2022)*. <https://doi.org/10.1109/CEC55065.2022.9870280>
- Mutsaddi, A., and Choudhary, A., 2025. Enhancing plagiarism detection in Marathi with a weighted ensemble of TF-IDF and BERT embeddings for low-resource language processing. *Proceedings of the International Conference on Computational Linguistics (COLING)*, pp. 89–100. <https://aclanthology.org/2025.loreslm-1.6/>
- Nahian, J.A., Srabon, M.M., Noori, S.R.H., and Masum, A.K.M., 2022. Review on multiple plagiarism: A performance comparison study. *Proceedings of the 13th International Conference on Computing Communication and Networking Technologies (ICCCNT 2022)*. <https://doi.org/10.1109/ICCCNT54827.2022.9984577>
- Paiva, J.C., Leal, J.P., and Figueira, Á., 2025. Clustering source code from automated assessment of programming assignments. *International Journal of Data Science and Analytics*, 20(2), pp. 1581–1592. <https://doi.org/10.1007/s41060-024-00554-5>.
- Potthast, M., Barrón-Cedeño, A., Stein, B., and Rosso, P., 2011. Cross-language plagiarism detection. *Language Resources and Evaluation*, 45(1), pp. 45–62. <https://doi.org/10.1007/s10579-009-9114-z>
- Saglam, T., Brodel, M., Schmid, L., and Hahner, S., 2024. Detecting automatic software plagiarism via token sequence normalization. *Proceedings of the International Conference on Software Engineering*, pp. 1384–1396. <https://doi.org/10.1145/3597503.3639192>
- Sağlam, T., Hahner, S., Wittler, J.W., and Kühn, T., 2022. Token-based plagiarism detection for metamodels. *Proceedings of the ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2022 Companion)*, pp. 138–141. <https://doi.org/10.1145/3550356.3556508>
- Sajid, M., Sanaullah, M., Fuzail, M., Malik, T.S., and Shuhidan, S.M., 2025. Comparative analysis of text-based plagiarism detection techniques. *PLoS ONE*, 20(4), pp. 1–28. <https://doi.org/10.1371/journal.pone.0319551>.
- Saqabi, A.A., Stewart, C., Akrida, E., and Cristea, A.I., 2025. A deep learning approach for paragraph-level paraphrase generation for plagiarism detection. *Neural Processing Letters*, 57(3), pp. 1–42. <https://doi.org/10.1007/s11063-025-11771-9>.
- Setu, D.M., 2025. A comprehensive strategy for identifying plagiarism in academic submissions. *Journal of Umm Al-Qura University for Engineering and Architecture*, 16(2), pp. 310–325. <https://doi.org/10.1007/s43995-025-00108-1>.
- Shikshan Sanshodhan, 2019. *Gujarati informative article, Paper ID: SS201906004. Shikshan Sanshodhan: Journal of Arts, Humanities and Social Sciences*.
- Veisi, H., Golchinpour, M., Salehi, M., and Gharavi, E., 2022. Multi-level text document similarity estimation and its application for plagiarism detection. *Iran Journal of Computer Science*, 5(2), pp. 143–155. <https://doi.org/10.1007/s42044-022-00098-6>



- Wahle, J.P., Ruas, T., Foltýnek, T., Meuschke, N., and Gipp, B., 2022. Identifying machine-paraphrased plagiarism. *Lecture Notes in Computer Science*, 13192, pp. 393–413. https://doi.org/10.1007/978-3-030-96957-8_34
- Wahle, J.P., Ruas, T., Kirstein, F., and Gipp, B., 2022. How large language models are transforming machine-paraphrased plagiarism. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*, pp. 952–963. <https://doi.org/10.18653/v1/2022.emnlp-main.62>
- Yalcin, K., Cicekli, I., and Ercan, G., 2022. An external plagiarism detection system based on part-of-speech (POS) tag n-grams and word embedding. *Expert Systems with Applications*, 197. <https://doi.org/10.1016/j.eswa.2022.116677>
- Zahid, M.M., 2023. An efficient machine learning approach for plagiarism detection in text documents. *Journal of Computing and Biomedical Informatics*, 4(2), pp. 241–248. <https://www.jcbi.org/index.php/Main/article/view/153>
- Zimba, O., and Gasparyan, A.Y., 2021. Plagiarism detection and prevention: A primer for researchers. *Reumatologia*, 59(3), pp. 132–137. <https://doi.org/10.5114/reum.2021.105974>
- Zouaoui, S., and Rezeg, K., 2022. Multi-agents indexing system (MAIS) for plagiarism detection. *Journal of King Saud University - Computer and Information Sciences*, 34(5), pp. 2131–2140. <https://doi.org/10.1016/j.jksuci.2020.06.009>

إطار عصبي دلالي عميق لاكتشاف الانتحال في الوثائق الغوجاراتية: GPlagNet

بومي شاه¹، *، غوراف كومار أميتا²

¹ قسم علوم وهندسة الحاسوب، كلية الهندسة والتكنولوجيا، جامعة بارول، واغوديا، فادودارا، غوجارات، 391760، الهند.

² قسم علوم وهندسة الحاسوب، معهد بارول للتكنولوجيا، كلية الهندسة والتكنولوجيا، جامعة بارول، واغوديا، فادودارا، غوجارات، 391760، الهند.

الخلاصة

للا تزال مشكلة اكتشاف الانتحال في اللغات منخفضة الموارد تمثل تحديًا مهمًا بسبب ندرة الموارد اللغوية وعدم قدرة أساليب المطابقة المعجمية التقليدية على اكتشاف التشابه الدلالي. وتُعد اللغة الغوجاراتية إحدى اللغات الهندية الرئيسية، إلا أنه لا يوجد حتى الآن نظام قوي لاكتشاف الانتحال قادر على تحديد الوثائق المعاد صياغتها أو المتشابهة دلاليًا ضمن السياق نفسه. في هذا البحث، يقدم المؤلفون GPlagNet، وهو إطار عصبي دلالي عميق لاكتشاف الانتحال في الوثائق الغوجاراتية، يدمج المعالجة المسبقة الخاصة باللغة مع التعلم القائم على تمثيلات المحولات (Transformers) يدمج الإطار المقترح عمليات التطبيع الخاصة باللغة، وتقسيم النص إلى رموز (Tokenization)، وإزالة الكلمات الشائعة (Stop-word Removal)، والتجذير (Stemming)، والإرجاع إلى الصيغة الأساسية للكلمات (Lemmatization)، قبل إنشاء تمثيلات سياقية للوثائق باستخدام محول ALBERT. وتنتج بنية المرمز الثنائي (Bi-Encoder) تمثيلات ثابتة الطول للوثائق، والتي تتم مقارنتها باستخدام تشابه جيب التمام (Cosine Similarity) لحساب درجات التشابه الدلالي وترتيب الوثائق المرشحة باستخدام استراتيجية الاسترجاع Top-K. حقق الإطار المقترح درجة 1.00 للوثيقة المتطابقة، ودرجات تشابه دلالي بلغت 0.8054 و0.8606 للوثائق المختلفة معجميًا ولكن المرتبطة سياقيًا، مما يشير إلى قدرة النموذج على التقاط العلاقات الدلالية التي تتجاوز مجرد التداخل المعجمي. وتُظهر النتائج أن GPlagNet تمكن من تحديد العديد من الوثائق المرتبطة دلاليًا في ظل ظروف إعادة الصياغة الممثلة في مجموعة بيانات التقييم، مما يشير إلى إمكاناته كنهج واعد لدعم النزاهة الأكاديمية وتقييم تشابه الوثائق في اللغات منخفضة الموارد.

الكلمات المفتاحية: الكشف عن الانتحال في اللغة الغوجاراتية، التعلم الدلالي العميق، ALBERT، تشابه المستندات، شبكات Bi-Encoder.