A Project Report
On

Multilingual (Hindi-English) Text Similarity Identification and Analysis using Semantic approach

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Degree of

Bachelor of Technology
In
Information Technology

Submitted By
Durgesh Bhadane
Janhavi Jadhav
Sharayu Mane
Vandana Pabale

Supervisor
Dr.Madhu Nashipudimath

Department of Information Technology
Pillai College of Engineering, New Panvel – 410 206

UNIVERSITY OF MUMBAI
Academic Year 2022– 23

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

CERTIFICATE

This is to certify that the requirements for the project report entitled '**Multilingual (Hindi-English) Text Similarity Identification and Analysis using Semantic approach'** have been successfully completed by the following students:

Name	**Roll No.**
Durgesh Bhadane	B802
Janhavi Jadhav	B805
Sharayu Mane	B809
Vandana Pabale	B823

In partial fulfillment of Bachelor of Technology of Mumbai University in the Department of Information Technology, Pillai College of Engineering, New Panvel – 410 206 during the Academic Year 2022 – 2023 .

_____

Supervisor

Dr.Madhu Nashipudimath

_____ _____

Head, Department of Information Technology **Principal**

(Dr. Satishkumar Varma) **(Dr. Sandeep M. Joshi)**

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

PROJECT APPROVAL FOR B.E

This project entitled "**Multilingual (Hindi-English) Text Similarity Identification and Analysis using Semantic approach**" by **Durgesh Bhadane, Janhavi Jadhav, Sharayu Mane, and Vandana Pabale** are approved for the degree of Bachelor of Technology in Information Technology.

Examiners:

1. _____

2. _____

Supervisors:

1. _____

2. _____

Chairman:

1. _____

Date:

Place:

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

DECLARATION

We declare that this written submission for B.E project entitled "**Multilingual (Hindi-English) Text Similarity Identification and Analysis using Semantic Approach**" represent our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

Project Group Members:

Durgesh Bhadane: _____

Janhavi Jadhav: _____

Sharayu Mane: _____

Vandana Pabale: _____

Date:

Place:

Table of Contents

Abstract

With the extensive amount of information in multiple languages available on the Internet, People tend to change the language of the previously published paper and publish it again using different languages. This cannot be detected using plagiarism software, detecting cross-lingual text reuse is a strenuous task. A multilingual system will be proposed to detect text similarity between Hindi-English language pairs using a semantic approach to overcome this enigma. The proposed model follows the steps- A sample Hindi document will be translated into English with the help of Google API. Further, Pre-processing will be done on the translated English document by applying filtration, stop word removal, and lemmatization. A sample Hindi document will be translated into English with the help of Google API. Further, Preprocessing will be done on the translated English document by applying filtration, stop word removal, and lemmatization.This data will be further pushed to the semantic algorithms.The similarity score will be computed with the help of a semantic approach. It will include the Word2Vector algorithm and Long-Short Term Memory(LSTM) algorithm. The final output will display the similarity score of both algorithms that can define which model is more efficient. The similarity score will be computed with the help of a semantic approach. It will include the Word2Vector algorithm and Long-Short Term Memory(LSTM) algorithm. The final output will display the similarity score of both algorithms that can define which model is more efficient.

List of Figures

List of Equations

List of Tables

Chapter 1

INTRODUCTION

Presently, with the broadening of worldwide networks, the volume of text data generated by humans is expanding. In the text data due to the proficiency of text alteration, similar text data is produced in abundance. As a consequence , plagiarism is an ethical issue for authors of text documents. Text-related research and applications, such as information retrieval, text categorization, and document clustering, are becoming more and more dependent on text similarity measurements.

Despite the fact that there are more than 7,000 languages spoken worldwide, only 23 are spoken by more than half of all people. Although it is challenging to identify duplication and text resemblance within the same dialect owing to the increasing utilization of data-driven approaches, it is even more challenging to recognize these issues while comparing texts in other dialects. The primary issue arises when previously published material is plagiarized and thereafter republished using a distinct language, which renders it difficult for algorithms and plagiarism detection software to spot the duplicated work.

Words can be related to one another both lexically and semantically.If the character order of two words is identical, then lexical similarity between the words exists. When two words have the same meaning, contrast one another, are used in the same way, are used in the same context, or are one type of another, semantic similarity between the words exists. Our primary goal is to advance the field of study by offering semantic approaches to multilingual text similarity that will increase its accuracy. This will assist in preserving the uniqueness of the work and in identifying any instances of plagiarism. A measure used to compare two texts and show their similarity is called text similarity. Two bits of text are compared for surface similarity or for similarity in meaning to calculate text similarity. Semantic similarity compares the meanings of two sentences to determine their similarity, while lexical similarity examines the surface closeness of the sentences. If two phrases have the same word set, the lexical similarity test considers them to be very similar and almost identical. Although two phrases may have similar word sets, semantic similarity can be radically different if the phrases have different meanings. Determining the degree of similarity between two texts is a critical and necessary task in many

information retrieval applications. A crucial and required task in many information retrieval applications is determining how closely two texts resemble one another. The effectiveness of numerous natural language processing (NLP) applications, including text summarization, machine translation, plagiarism detection, sentiment analysis, etc., depends on the textual and semantic similarity. Additionally, it is crucial to calculate the similarity score between two sentences that are written in different languages.

1.1 Fundamentals

Text Similarity is the process of comparing a piece of text with another and finding the similarity between them. It's basically about determining the degree of closeness of the text. Dealing with text, sentences or words brings us to the region of Natural Language Processing (NLP), where we are going to use different NLP approaches to process the raw text and help the model to detect the similarity more swiftly and efficiently. To execute this task, the input text must be converted into a more machine-readable form by transforming the text into embeddings which then get converted into vectors that are understood by the machine to calculate the similarity.

Semantic similarity refers to the similarity of two pieces of text when their contextual meaning is considered. It judges the order of occurrences and the meanings of the words in the text. Semantic similarity is often used to address NLP tasks such as paraphrase identification, automatic question answering and removing similar sentences(redundancy removal).

1.2 Objectives

The main objective Semantic Similarity is to measure the distance between the semantic meanings of a pair of words, phrases, sentences, or documents.

- Prevent plagiarism
- Check the similarity accuracy of semantic approach
- Maintaining the originality of the work done by author
- Visualization of similarity accuracy LSTM and Word2Vec algorithms

1.3 Scope

Text similarity is one of the active research and application topics in Natural Language Processing. With the increasing popularity of the Internet in various parts of the world, the languages used for Web documents have expanded from English to various languages. However, there are many unsolved problems in order to realize an information system which can handle such multilingual documents in a unified manner.

The Internet Archive archives sites on the web, and has reached the size of approximately 1 petabyte of data and is currently growing at a rate of 20 terabytes per month. With such a large amount of text, English and non-English alike, it is difficult to filter and manage the information that people need between all this there can be probably a chance where most of the multilingual documents are similar and existing systems are trying their best to make this process more automated by removing the multilingual plagiarized work from the internet. This task can only be done using multilingual text similarity.

1.4 Organization of the Report

The report is organized as follows: The introduction to the text similarity is given in Chapter 1 It describes the types of similarity . This chapter also presents the outline of the objective of the report. Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the previous work done on the related topic. Chapter 3 presents the implementation of the proposed work. It describes the major approaches used in this work. The techniques are explained which are being used for the project. The societal and technical applications are mentioned in Chapter 4. The summary of the report is presented in Chapter 5.

Chapter 2

Literature Survey

2.1 Introduction

In this chapter the relevant techniques in literature are reviewed. This literature survey aims to explore and analyze the existing research in multilingual text similarity identification using semantic approaches. The survey will cover studies that have utilized various algorithms for measuring text similarity between multilingual documents. The survey will also highlight the strengths and limitations of these approaches, identify research gaps and challenges, and suggest future research directions in this area.

2.2 Literature Review

[1]. Direct and pivot translation probabilities are examples of translation-based similarity measurements.Analysis of these metrics inside the general multilingual NMT framework. The easiest measurement is direct translation likelihood. Pivot translation probability is useful, especially when comparing languages because it regards inputs as target sequences.

[2]. Complete sentences and their semantic information are represented as vectors via sentence embedding techniques. The context,semantics and other subtle features of the sentence can be represented in these embeddings.to calculate the semantic similarity, various custom-made bag-of-words based approaches are built using word embedding techniques such as Word2vec and GloVe.

[3]. The two linguistic sub-branches of semantics and morphology were taken into consideration when clustering the word vectors from the Ukrainian corpus. First, it was looked into how well the words describe the clusters to which they belong and how the vectors are derived from the Ukrainian corpus. Second, the relationship between word vector clustering and the morphological characteristics of Ukrainian suffixes was examined.

[4]. Traditional statistical techniques, deep learning models, and innovative models built on several models were combined to address the issue. The part-of-speech-based Jaccard coefficients are used in the models. For phrase similarity, the Term Frequency-Inverse Document Frequency (TF-IDF) and word2vec-CNN algorithms are used.

[5]. For this inquiry, the Word2Vec model is developed in a number of different ways to find the best similarity value. Configuration is done by adjusting the window size settings as well as the word vector dimensions.

[6]. Currently, NLP systems have various limitations. To make the leap forward from natural language processing to natural language understanding, search needs to focus on semantically related concepts that enable performing complex NLP tasks. This study attempts to contribute to a paradigm shift from natural language processing to natural language comprehension by focusing on semantic similarities across texts in domain-specific settings.

[7]. Siamese CNN is used to analyze the local context of words in sentences. This algorithm also creates visuals of word neighborhoods and relevance. A Siamese LSTM is then used to parse the entire phrase based on the words and the local context. Finally, use the Manhattan distance to calculate the semantic similarity between two sentences

[8]. Present a Siamese adaptation of long short-term memory (LSTM) networks to tagged data consisting of pairs of variable-length sequences. It provides the LSTM with word embedding vectors supplemented with synonym information that encodes the underlying meaning expressed in the sentence using fixed-size vectors (regardless of the particular formulation/syntax).

2.3 Summary of Literature Survey

Table 1.Summary of Literature Survey

Sr. No	Literature	Feature	Advantage	Constraints/ Future Scope
1	Rico Sennrich et al. 2022 [1]	● The theoretical and empirical characteristics of text similarity measures based on translation.	● Requires one translation direction to achieve symmetry and that the input languages are not required to be specified.	● The longest sequence supported by NMT models is frequently quite short.

2	Derry Jatnikaa et al.2019 [3]]	• Words are represented as vectors. • The 320,000 data points used to create the model	• Higher correlation and similarity values	• The training of the Word2Vec model requires a huge corpus,
3	Larysa Savytskaa et al. 2021 [6]	• Clustering of word vectors from a Ukrainian corpus. • Two linguistic subfields: semantics and morphology.	• Machine tech enables computerizing language analysis, Word2vec is helpful	• In-depth analysis to compute linguistics concepts • Creating semantic maps and expanding inquiries.
4	Md Shajalal, Masaki Aono 2018[10]	• Trials on a dataset for semantic textual similarity for Bengali texts to evaluate the effectiveness of our strategy.	• Computed the word-level semantics using a pre-trained word-embedding model	• Plan of using Long Short-Term Memory (LSTM)
5	Peiying Zhang et al. 2021 [12]	• A multi feature weighting mechanism is added to the word2vec CNN model based on multiple features.	• The weighting mechanism can highlight the key points of extraction.	• The word vector given by the word2vec model is static and cannot describe the dynamic change of semantics.
6	Reddy Guddeti et al.2021 [13]	• Examination on the extraction of embeddings from text	• Novel model performs better than other models. • It can efficiently extract semantic information from the input text.	• Assessing the degree of similarity between groups of images embedded in various publications.
7	Surabhi Som 2019 [22]]	• Semantic similarity attempts to determine the degree of similarity between words using data produced from semantic networks rather than data from huge corpora.	• Ontology, taxonomies, and semantic internet aid in data retrieval.	• Stemming and lemmatization can be applied to improve sentence similarity classification

8	Elvys Linhares Pontes e.t. al.2018 [23]	• Combines convolution and recurrent neural networks	• The analysis of general and local contexts of words improved the sentence analysis.	• Involve exploring the model's generalization capability to other NLP tasks
9	Jonas Mueller et al.2016 [26]	• LSTM is used to encode the underlying meaning of a sentence using a fixed-length vector	• Performs better than more complex neural network systems and state-of-the-art features	• Testing the model on different datasets for other languages and evaluating its performance

Chapter 3

Multilingual Text Similarity and Analysis using Semantic Approach

3.1 Overview

Multilingual (English – Hindi) text similarity identification using semantic approach will allow users to check whether two text documents of different languages (English-Hindi) are indistinguishable or not. To compare both languages semantically, Word2Vec and LSTM algorithms are used. Multilingual text similarity identification between English and Hindi can be done using semantic approaches like Word2Vec and LSTM algorithms. Word2Vec uses neural networks to learn vector representations for words, which can be used to compute similarity between words in both languages. LSTM is a recurrent neural network that can learn language-agnostic representations of text for comparison between documents in different languages. These approaches can provide a powerful tool for measuring text similarity in a multilingual setting.

1. Firstly, the Hindi text document will be sent to the language translation API.
2. After that the language translation API will convert the Hindi text into English text.
3. Once the translation is done then the translated English text will be used for pre-processing
4. Filtration.
5. Stop word removal.
6. Lemmatization.
7. Semantic analysis using Word2Vec and LSTM algorithms for comparing the hindi and english text document.
8. Comparing the outputs of algorithms using Cosine similarity and Pearson correlation.
9. Result analysis using visualization.

3.1.1 Existing System Architecture

Figure 3.1 The Existing system takes input as a set of English text documents. The documents undergo pre-processing steps which include tokenization, stop word removal, lemmatization, and POS tagging. Then the pre-processed data is given as input to compute semantic similarity for plagiarism detection. The final output is a report produced based on a set threshold value which decides whether both the documents have similar content or not. The existing system focuses on monolingual semantic plagiarism detection.

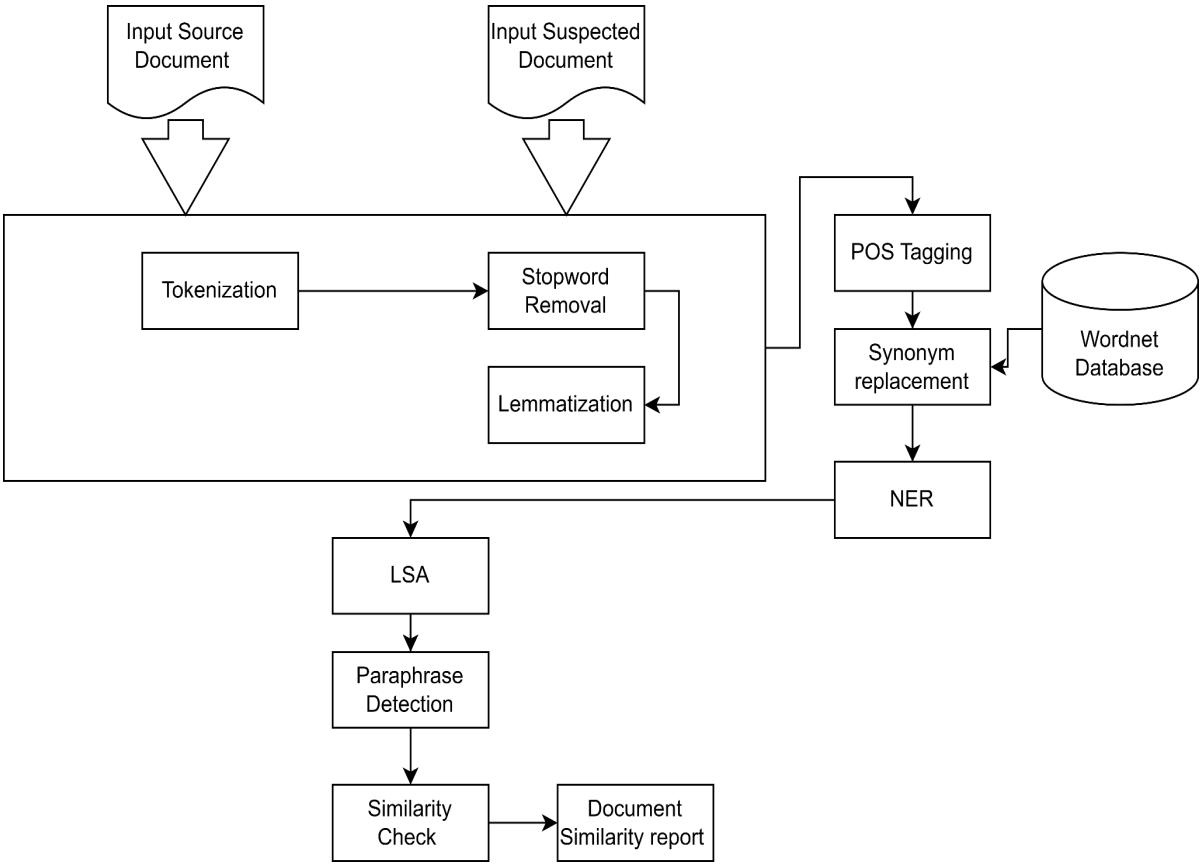

Figure 3.1 Existing system architecture

The existing system for finding semantic similarity is limited to the English language and does not offer support for other languages. This is a disadvantage, as static and semantic similarity measures tend to produce similar results when applied to text within the same language.

3.1.2 Proposed System Architecture

Figure 3.2 shows the proposed system architecture. A Hindi text document is used as input and translated into an English document. The resulting document will be preprocessed with filtration, stopword removal and lemmatization to get informative data which will be more understandable to the machine. further the fully processed document will be used as an input to both semantic algorithms LSTM and Word2Vec. Both algorithms will be tested by performing similarity scores of

Hindi and English text documents, the outputs of both algorithms will be compared. Finally, the detailed visualization of both algorithm outputs will be done.

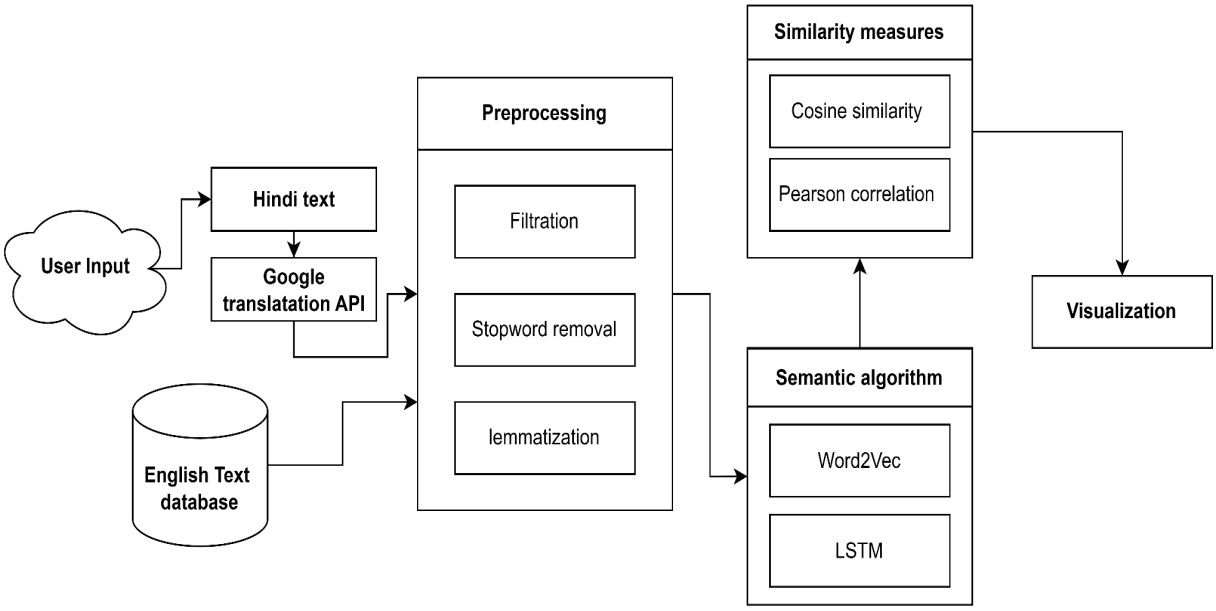

Figure 3.2 Proposed System Architecture

3.2. Implementation Details

3.2.1 Techniques and Methodologies

1. **Data Input:**

   Hindi text document:

   These are the documents which will contain text data of their respective languages.

2. **Google Translation API:**

   It is a Python library that provides an interface for Google Translate API to perform language translation in Python. It uses the Google Translate API to translate text between different languages. The library supports more than 100 languages and allows you to translate text, sentences, or even entire documents. This translated file is pre-processed using the steps followed for pre-processing of English data files.

3. **Filtration:**

   Filtration is a process to remove unnecessary data in the text like (, ! & * $ # @ . / ”:) , if we don’t remove this data while preprocessing then it will become noise further.

   Example Input:

   To become more successful in coding, solve more real problems for real people.That’s how you polish the skills you really need in practice. After all, what’s the use of learning theory that nobody ever needs?

   Filtered text:

To become more successful in coding solve more real problems for real people that's how you polish the skills you really need in practice after all what's the use of learning theory that nobody ever needs

4. **Stop word in English**:

Stop words are a set of commonly used words in a language. Examples of stop words in English are "a", "the", "is", "are" and etc. By removing these words, we remove the low-level information from our text in order to give more focus to the important information.

Example Input:

To become more successful in coding solve more real problems for real people that's how you polish the skills you really need in practice after all what's the use of learning theory that nobody ever needs.

Tokenization:

['to', 'become', 'more', 'successful', 'in', 'coding', 'solve', 'more', 'real', 'problems', 'for','real', 'people', 'that's', 'how', 'you', 'polish', 'the', 'skills','you','really', 'need', 'in', 'practice', 'after', 'all', 'what's', 'the', 'use', 'of', 'learning', 'theory', 'that', 'nobody', 'ever', 'needs']

Output:

Stopwords recognized in the given sentence: ['that', 'of', 'you', 'after', 'to', 'the', 'what','in', 'for', 'how', 'all', 'more']After removing the recognized stopwords, the Tokens of sentence is: ['become','successful', 'coding', 'solve', 'real', 'problems', 'real', 'people', 'polish', 'skills','really','need', 'practice', 'use', 'learning', 'theory', 'nobody', 'ever', 'needs']

5. **Lemmatization:**

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings and to return the base or dictionary form of a word, which is known as the lemma. If confronted with the token saw, stemming might return just s, whereas lemmatization would attempt to return either see or saw depending on whether the use of the token was as a verb or a noun.

Example:

car, cars, car's, cars' = car

6. **Semantic algorithms:**

   Word2vec and LSTM algorithms will be used for getting the similarity score of the document and each of these algorithms will show the accuracy of the text similarity[6] . This similarity score will decide how much the Hindi document is similar to the English document.

   For example: if the similarity score of a document is 15%, then 15% of the content written is unoriginal.

7. **Similarity Measures:**

- **Cosine similarity-**

  Cosine similarity is a commonly used technique in natural language processing for measuring the similarity between two pieces of text. It is a measure of the cosine angle between two vectors representing the two pieces of text in a high-dimensional space.

- **Pearson Correlation coefficient-**

  The Pearson correlation coefficient is a statistical measure that assesses the linear relationship between two variables. In the context of text similarity, it can be used to measure the degree of similarity between two text documents.

8. **Visualization:**

   Visualization of the output will be done after performing all the processes and methods. This will help in understanding the output in a straightforward and briefly described way. A variety of python libraries are present which will show best visualization of the final output.

3.2.2 Algorithms

1. **Word2Vec:**

   In the field of Natural Language Processing, Word2Vec is a popular technique for creating word embeddings (NLP). Word2Vec generates word vectors, which are dispersed numerical representations of word properties. Some of these vectors may contain words that reflect the context of the particular vocabulary words that are individually present. By using the final produced vectors, word embeddings assist in establishing the relationship between a word and another word with similar meaning. Without requiring human input, Word2vec

generates vectors, which are distributed vector representations of the word properties or how words are used in context.

Word2Vec Algorithm steps:

1. Read the suspicious Hindi file and store its content in a variable.
2. Preprocess the Hindi text by removing stop words, tokenizing, and lemmatizing the tokens.
3. Translate the preprocessed Hindi text to English using a translation API or library.
4. Tokenize the translated English text.
5. Load the pre-trained Word2Vec model, which has been trained on a large English corpus.
6. Remove out-of-vocabulary tokens from the English tokens list.
7. Calculate the vector representation of the English text by taking the average of the word vectors of all the tokens in the text.
8. Read each English file that needs to be compared and store its content in a variable.
9. Tokenize the English text of each file.
10. Remove out-of-vocabulary tokens from the English tokens list.
11. Calculate the vector representation of the English text of each file by taking the average of the word vectors of all the tokens in the text.
12. Calculate the cosine similarity between the vector representation of the suspicious file and the vector representation of each English file.
13. Rank the English files based on the cosine similarity scores and return the top N most similar files.

Word2vec can offer very precise estimates or predictions about a word's meaning based on prior occurrences if given sufficient information, such as context. This Word2Vec model was built using training data from a big corpus of English 2 Wikipedia. The Python Gensim module is essential to this study since it provides all the features required to build a Word2Vec model [13].

Word2vec clusters resemble the syntactic and semantic relationships found in natural language. "Great," "greater," or "easy," "easiest" word pairs are examples of syntactic relationships between words in English. The Word2vec-generated vectors are clustered, just like the syntax relations in these word pairs.

The words ``Athens" and "Greece" are a pair of English words that have the same semantic meaning as "country" and "capital," respectively. Similarly to this, the words "King" and

"Queen," which are both terms of nobility, have a semantic relationship. Word vectors cluster according to the semantic connections between the words as a result of their interactions with one another. One can get logical outcomes if the word is represented correctly. By combining and subtracting the vectors obtained with Word2vec, a new vector can be created that produces semantic results based on cosine similarities [9].As an illustration of how arithmetic operations can yield semantic outcomes, consider the result vector acquired by changing the gender characteristic in the word "King," which indicates nobility.

For example(fig 3.5), words like “King” and “Queen” would be very similar to one another. When conducting algebraic operations on word embeddings you can find a close approximation of word similarities.

KING	-	MAN	+	WOMAN	=	QUEEN
[5,3]	-	[2,1]	+	[3,2]	=	[6,4]

Figure 3.5 Word2Vec Example

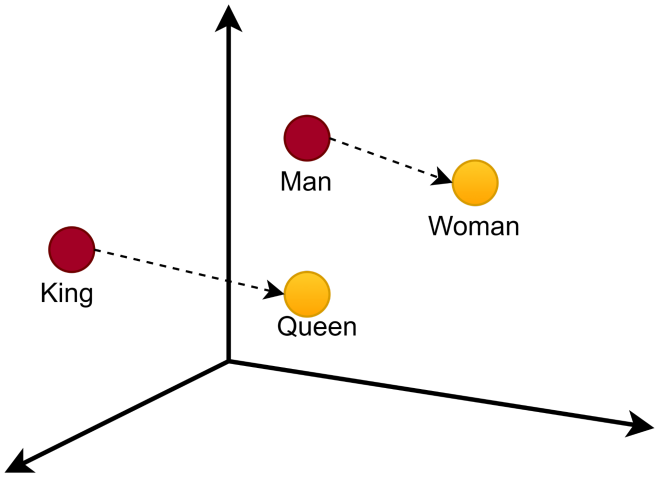

Figure 3.6 Word2Vec Representation

2. **LSTM(Long-Short term Memory):**

A long short term memory network is an enhanced RNN, a sequential network that allows information persistence. Address the vanishing gradient issue that RNN is experiencing. Each of the three components that make up the LSTM has a distinct purpose. The first section determines whether information from earlier timestamps should be remembered or if it is unnecessary and should be ignored. The cell then tries to extract fresh data from the input cell. Finally, the cell transmits the revised data from the present timestamp to the next timestamp. Gates refer to these three LSTM cell components. Forget Gate is the first section, Input Gate is the second, and Output Gate is the final section. Neural networks (NNs) use deep sentence and word analysis to better explain both the semantics and structure of sentences and predict sentence similarity.

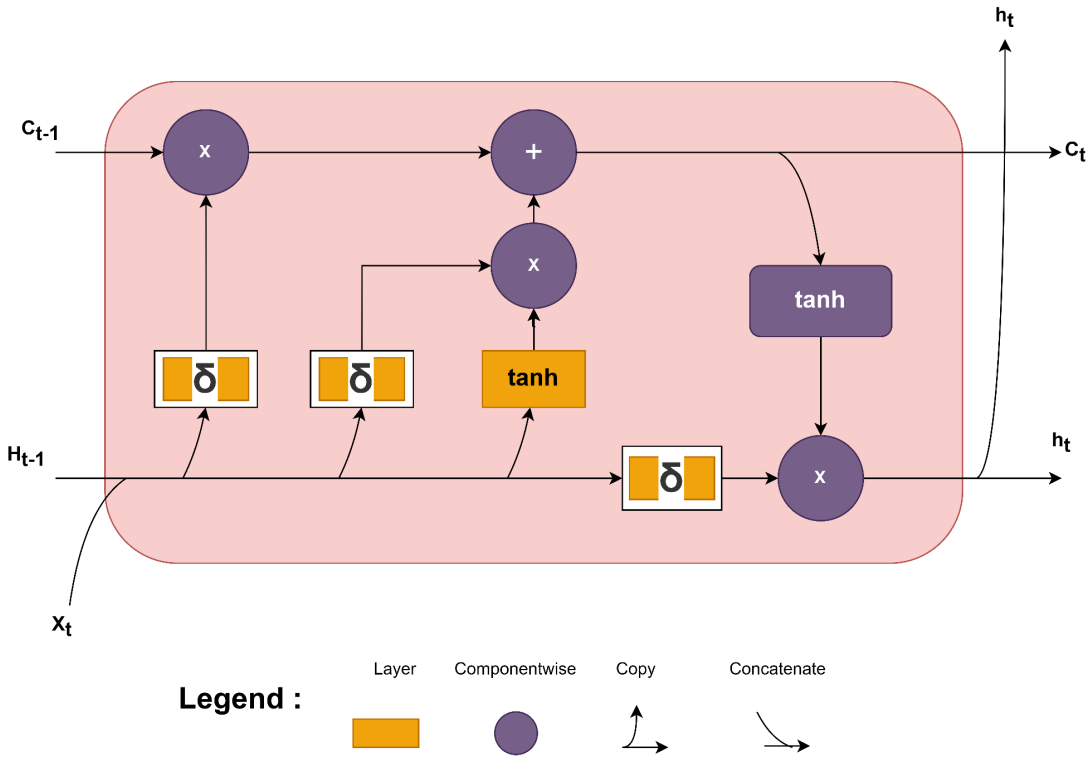

Figure 3.7 Working of LSTM

LSTM Algorithm steps:

1. Read the suspicious Hindi file from suspicious_file_path.
2. Translate the Hindi text to English using translate_text function.
3. Preprocess the translated text using preprocess_hindi function.
4. Tokenize the preprocessed Hindi text using nltk.word_tokenize function.

5. Encode the Hindi text using model1 LSTM model and extract the output embeddings.
6. Read the English file from english_file_path.
7. Tokenize the English text using the nltk.word_tokenize function.
8. Encode the English text using model1 LSTM model and extract the output embeddings.
9. Calculate cosine similarity between the Hindi and English embeddings.
10. Calculate Pearson correlation between the Hindi and English embeddings.
11. Return the cosine similarity and Pearson correlation as output.

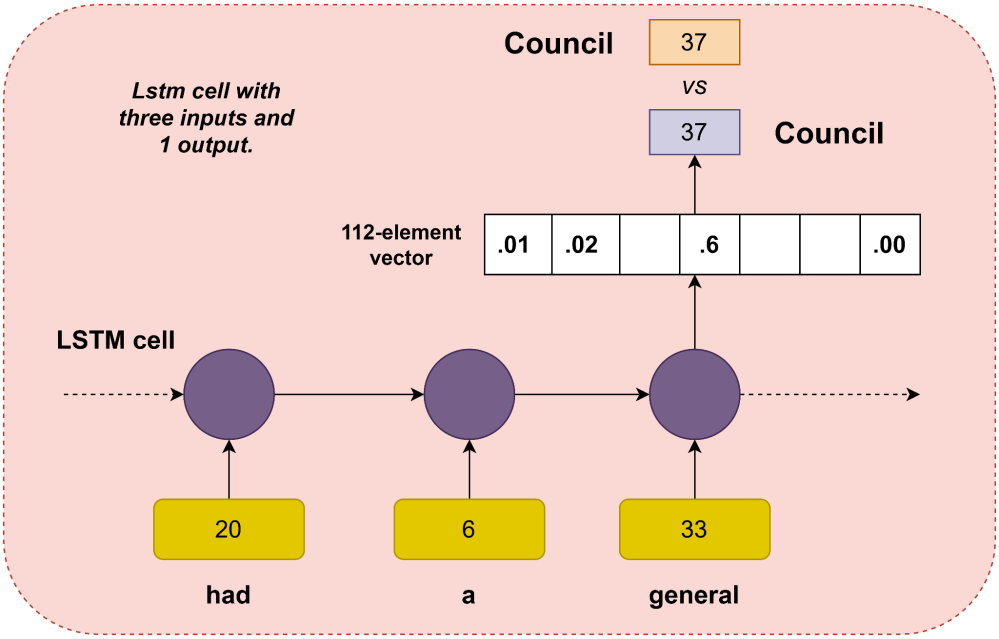

Figure 3.8 Example of LSTM

NN-based similarity measurement method. First, is Siamese CNN to analyze the local context of words in a sentence and to generate a representation of the relevance of a word and its neighborhood. Then, Siamese LSTM analyzes the entire sentence based on its words and its local contexts. Long Short Term Memory (LSTM) enhances RNNs to handle long-term dependencies. LSTM technology consists of memory cells and non-direct gating devices. These devices update state over time and manage the input and output flow of the cell. Siamese LSTM for encoding sentences using pre-trained word embedding vectors. Using the same weights, a Siamese LSTM encoded sentences to create comparable sentence representations of related sentences. The Manhattan distance between the sentence representations was then used to forecast how closely two sentences will be related. Tree-LSTM is a generalization of LSTM for tree-structured network topologies. This Tree-LSTM is used to encode some sentences and predict their proximity with an NN

that analyzes the distance and angle between sentence embeddings[19]. LSTMs outperform basic RNNs for long-range dependency learning because the use of memory cell units can store/access information over long input sequences. Like RNNs, the LSTM sequentially updates a hidden-state representation, but these steps also rely on a memory cell containing four components (which are real-valued vectors): an input (and forget) gate it (and ft) that regulates what is stored in (and omitted from) memory based on each new input and the current state, a memory state ct, an output gate ot that determines how the memory state impacts other units. [25]

3.2.3 Use case diagram

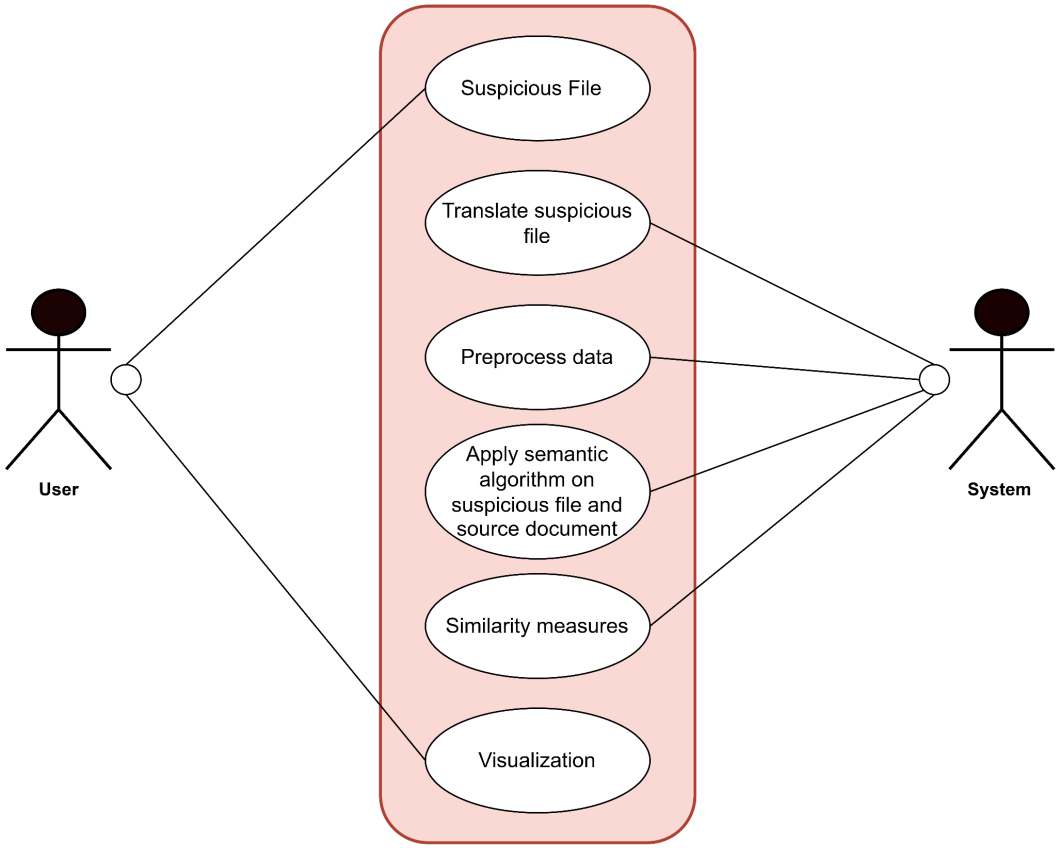

Figure 3.9 Use Case diagram

Fig3.9 use case diagram provides a high-level view of the interactions between the user and the multilingual text similarity system, and highlights the key functionality of the system in handling text input in multiple languages and using a semantic approach to calculate similarity. The user has two use cases: "Enter text to compare" and "View results." In the "Enter text to compare" use case, the user enters Hindi text to compare the similarity in English language. The system has four use cases: “translating suspicious file”, “preprocessing data”, “applying algorithms on the suspicious file”, “finding similarity measures”.

3.2.4 Hardware and Software Specifications hardware

The Hardware and Software requirements are stated as below:

Table 3.1 Hardware Details

Processor	I5 10th gen chip or above
RAM	8GB ddr3 or above

Table 3.2 Software Details

Operating System	Windows
Programming Language	Python
IDE	Pycharm

3.2.5 Applications

1. **Language translation:** Improving the accuracy of language translation. By comparing the similarity of texts in different languages, these tools can help to identify errors or inconsistencies in translations, and suggest revisions to improve the accuracy of the translation.
2. **Content creation:** Helping content creators to create original content by identifying instances where their writing is too similar to existing content. This can help to improve the quality of the content and prevent unintentional plagiarism.
3. **Plagiarism detection:** Detecting instances of plagiarism across different languages. This is particularly important in academic settings, where students may copy content from sources in languages other than their own.
4. **Machine learning:** Improving their accuracy in natural language processing tasks such as sentiment analysis, text classification, and language identification.
5. **Cross-cultural communication:** Bridging the gap between different languages and cultures, enabling effective communication and collaboration across language barriers.

Chapter 4

Result & Discussion

4.1 Input and Output screenshots

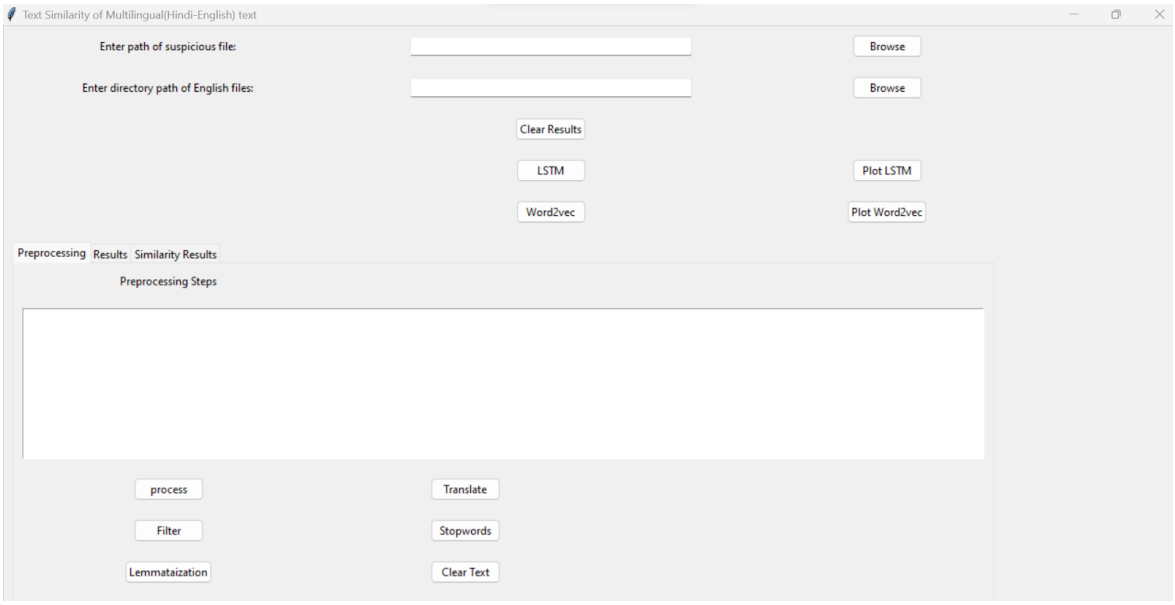

figure 4.1 GUI overview

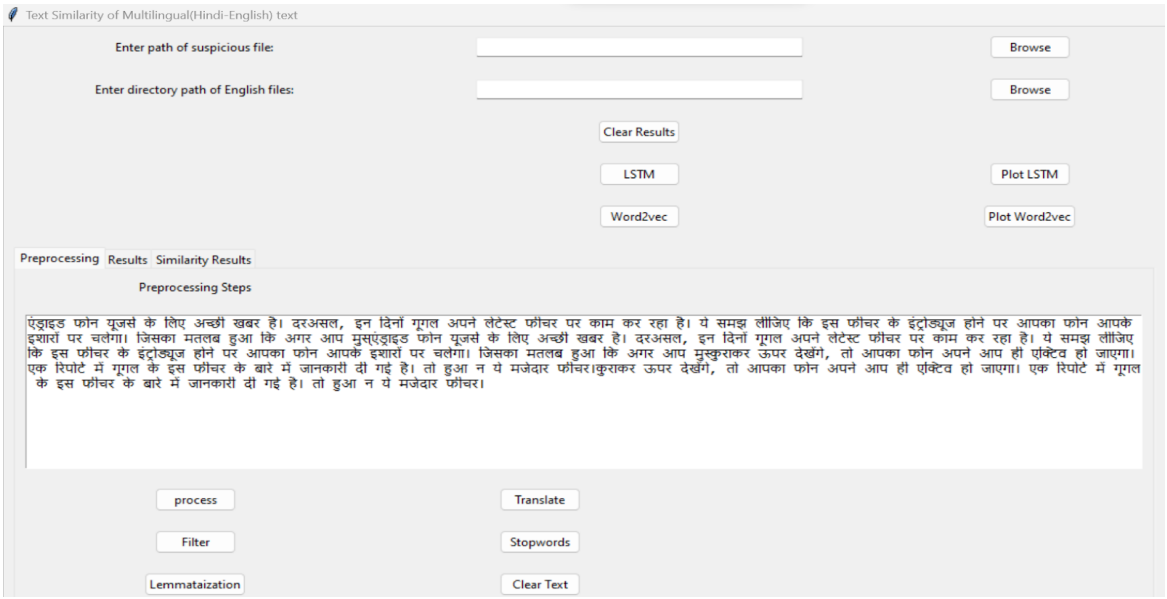

figure 4.2 Data input of hindi language to save in database

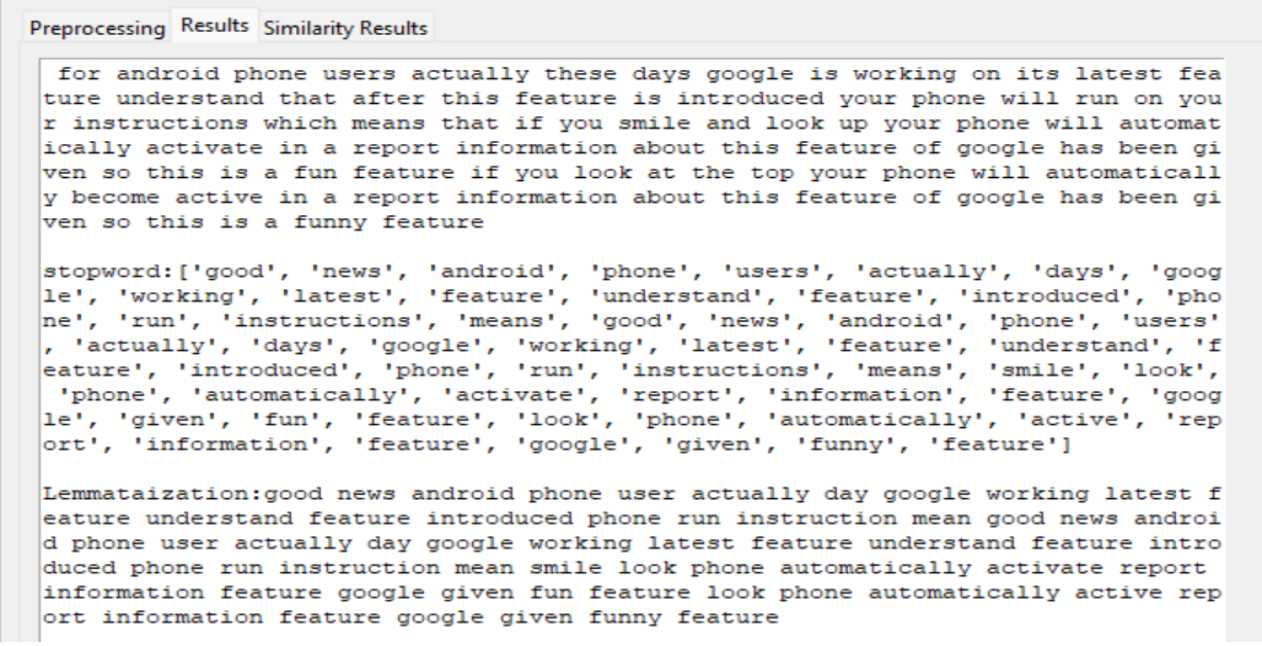

figure 4.3 Preprocessing of data

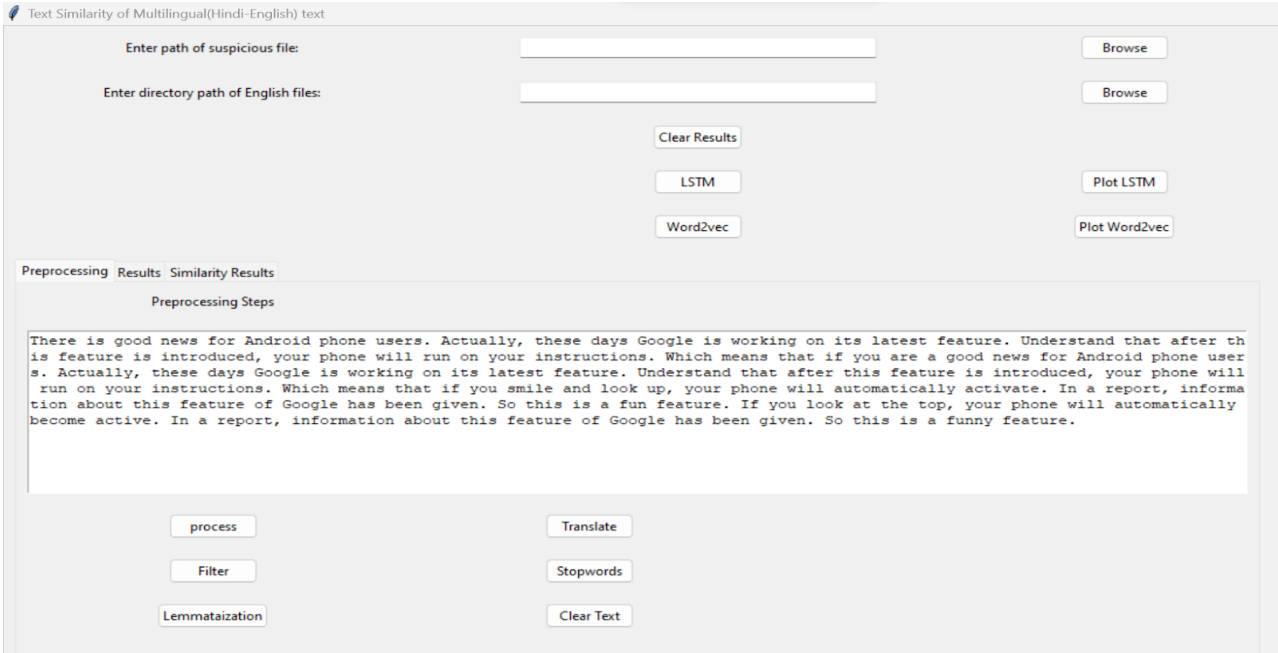

figure 4.4 English translated data from hindi text to save in the database.

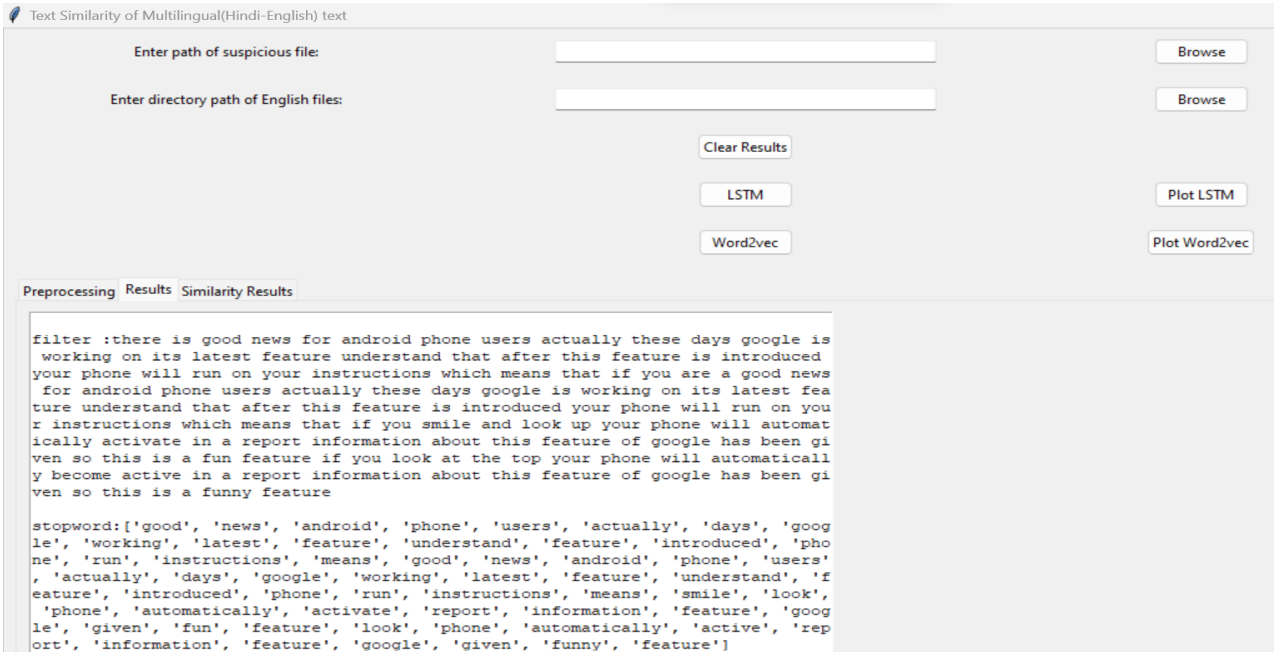

figure 4.5 English translated data preprocessing

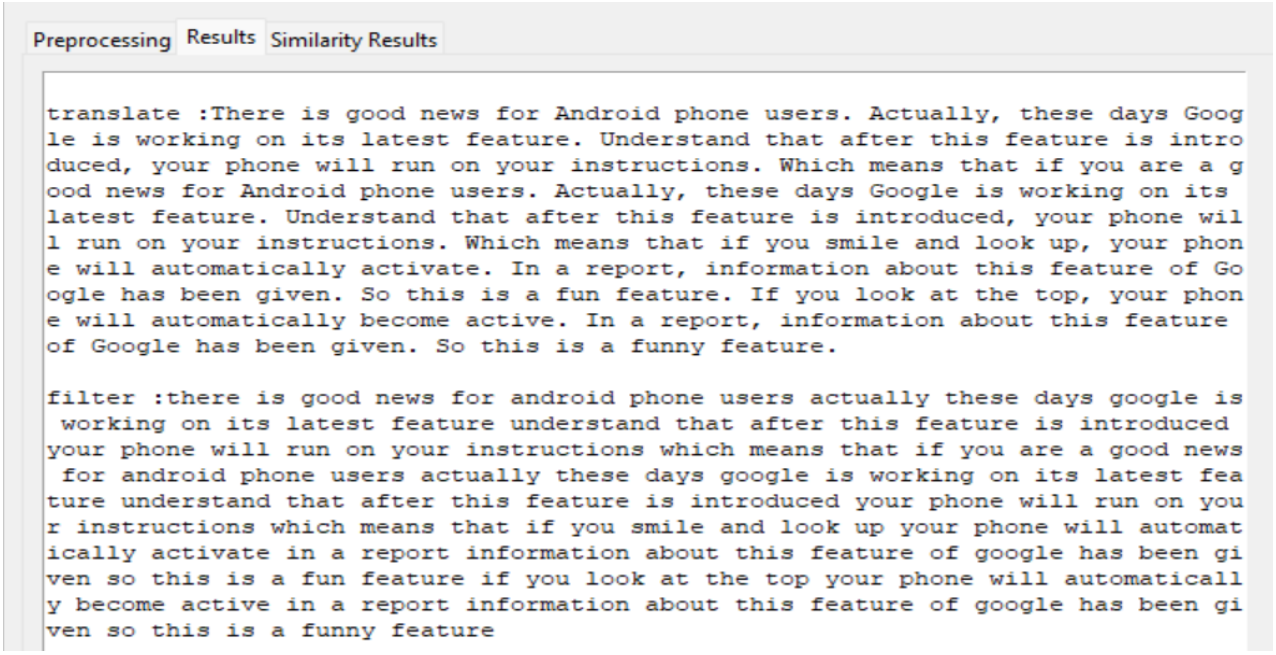

figure 4.6 English translated data preprocessing

4.2 Evaluation parameters

1.Cosine similarity:

The Word2Vec and LSTM model can be used to calculate the similarity between two documents. The basic idea is to first represent each document as a vector by taking the average of the word vectors of all the words in the document. Then, the similarity between the two document vectors is calculated using the cosine similarity measure.
The mathematical equation for computing the cosine similarity between two document vectors, d1 and d2, is as follows:

$$\textbf{cosine\_similarity}\ (d1, d2) = (d1 \cdot d2) / (||d1|| * ||d2||)$$

equation 4.3 Cosine similarity

where d1 and d2 are the

"||d1||" represents the Euclidean norm (magnitude) of d1

"||d2||" represents the Euclidean norm (magnitude) of d2

The dot product between two vectors measures the extent to which vectors point in the same direction, and the magnitudes of vectors measure their lengths. Therefore, cosine similarity takes into account both the direction and magnitude of vectors, making it a widely used measure for computing similarity between vectors.
In summary, the Word2Vec model can be used to represent documents as vectors, and the similarity between two documents can be calculated using the cosine similarity measure.document vectors." . " represents the dot product operation between d1 and d2.

2. Pearson Correlation Coefficient:

The Pearson correlation coefficient measures the linear relationship between two variables, typically denoted as X and Y. It ranges from -1 to 1, where a value of -1 indicates a perfect negative linear relationship, 0 indicates no linear relationship, and 1 indicates a perfect positive linear relationship.

$$\mathbf{r} = \frac{\mathbf{n}(\sum \mathbf{xy}) - (\sum \mathbf{x})(\sum \mathbf{y})}{\sqrt{[\mathbf{n}\sum \mathbf{x^2} - (\sum \mathbf{x})^2][\mathbf{n}\sum \mathbf{y^2} - (\sum \mathbf{y})^2]}}$$

equation 4.4 Pearson correlation coefficient

The Pearson correlation coefficient is computed using the pearson function from the scipy.stats module, which returns a tuple containing the Pearson correlation coefficient and the two-tailed p-value. The code indexes the first value of the tuple using [0] to obtain the Pearson correlation coefficient.

4.3 Performance evaluation

figure 4.7 Similarity result of LSTM using cosine similarity and pearson correlation coefficient

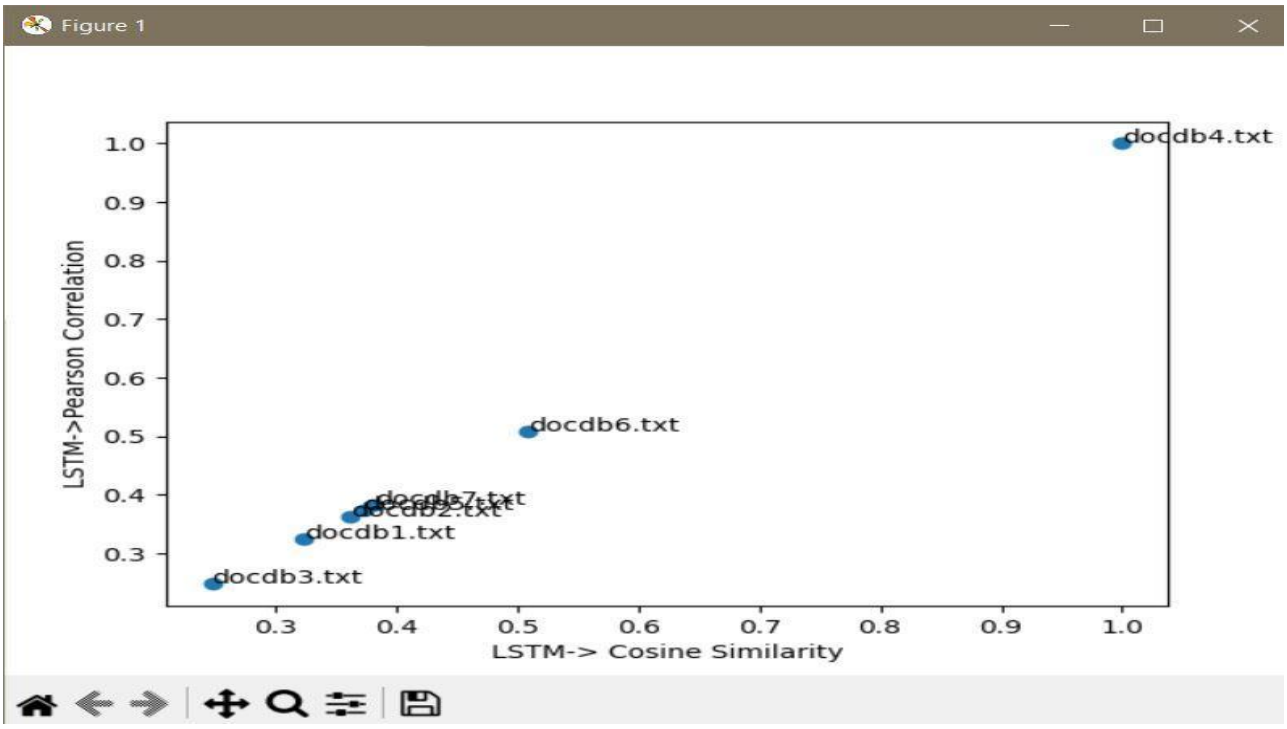

figure 4.8 LSTM graph representing According to the fig 4.7

figure 4.9 Similarity result of Word2Vec using cosine similarity and pearson correlation coefficient

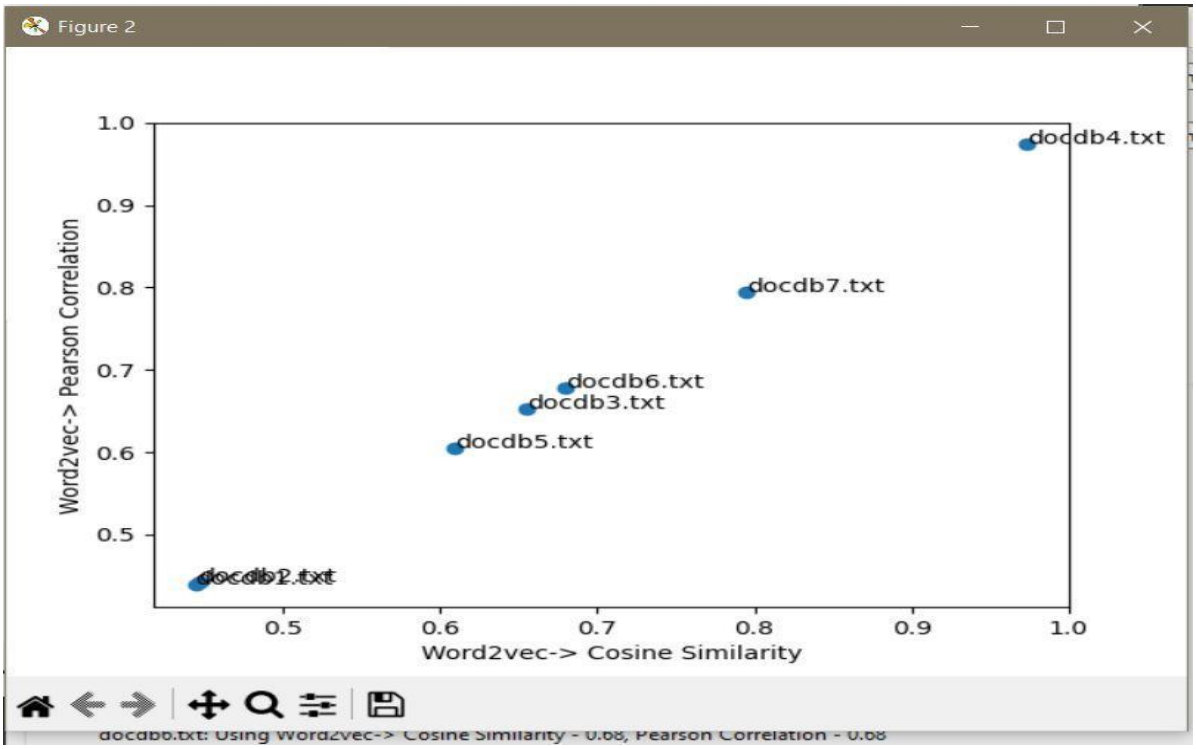

figure 4.10 Word2Vec graph representing According to the fig 4.9

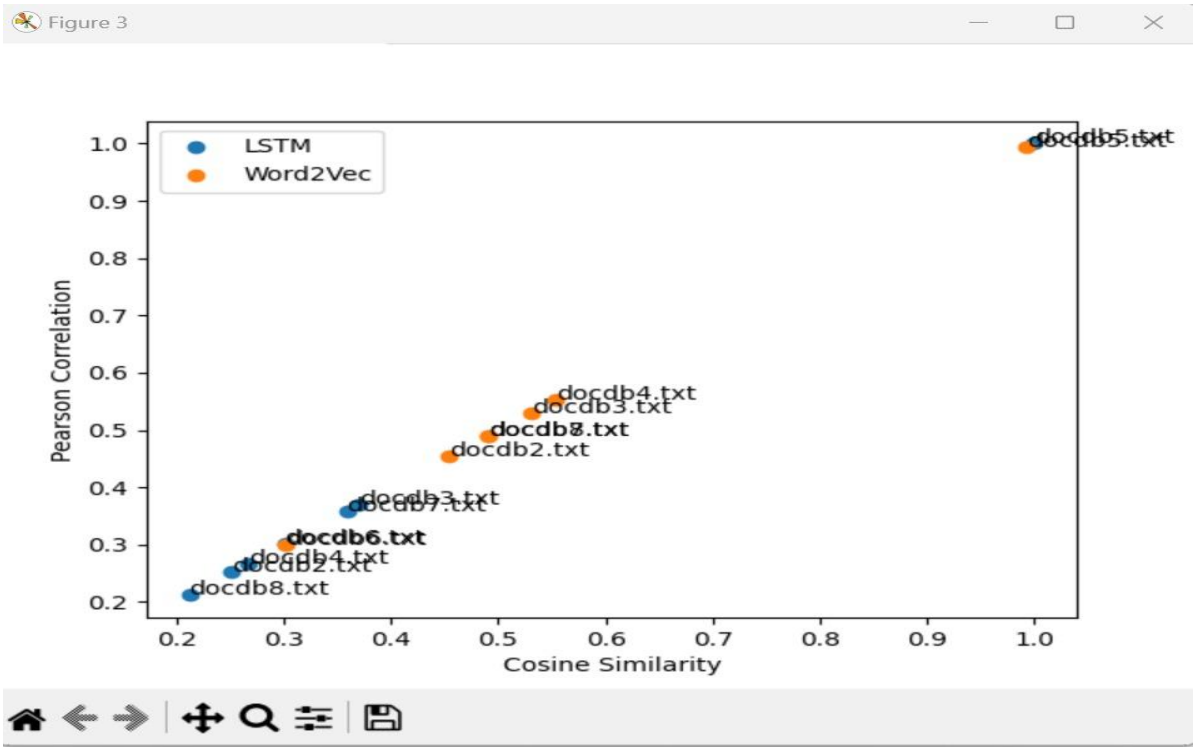

figure 4.11 Graphical Representation of LSTM and Word2Vec

Analysis using Scatter Plot. According to fig 4.9 and fig 4.7.

table 4.12 Comparison table of LSTM and Word2Vec algorithms.

	Word2Vec		**LSTM**	
Documents	Cosine Similarity	Pearson Correlation	Cosine Similarity	Pearson Correlation
Document 1	0.44	0.44	0.32	0.32
Document 2	0.45	0.44	0.36	0.36
Document 3	0.66	0.65	0.25	0.25
Document 4	0.97	0.97	1	1
Document 5	0.61	0.61	0.37	0.37
Document 6	0.68	0.68	0.51	0.51
Document 7	0.8	0.79	0.38	0.38

Table 4.12 provides a tabular representation of the comparison between LSTM and Word2Vec algorithms. Based on the results in Table 4.12 and Figure 4.11, the LSTM algorithm demonstrates better than the Word2Vec algorithm. Table 4.12 shows the similarity score of each algorithm with respect to Cosine similarity and Pearson correlation. A score of 1 indicates that the documents are completely identical, while a score close to 1 indicates that the documents may be similar, and score close to 0 indicates that the documents are dissimilar.

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

In this project The importance of text similarity and their applications has been explained. It has been observed that detecting text similarity is an important research problem which needs to be addressed with semantic approaches so as to improve its accuracy and performance. Also, it has been observed that there is less research done on **Multilingual Text similarity Identification** for Hindi-English language pairs. The semantic algorithms LSTM and Word2Vec were used to calculate similarity scores between Hindi and English documents. After comparing outputs of both the algorithms, the Lstm algorithm is found to be performing better than the Word2Vec algorithm for text similarity in hindi-english language pair.

5.2 Future Scope

1. **Real-time multilingual communication:** Integrating real-time communication tools like chatbots or virtual assistants. This would allow for seamless multilingual communication between individuals who speak different languages, without the need for a human translator.
2. **Multilingual content creation:** creating content in multiple languages simultaneously. This would enable content creators to reach a wider audience and produce content that is tailored to the specific cultural and linguistic nuances of different regions.
3. **Language learning:** facilitate language learning by comparing the similarity of texts in different languages. This would enable learners to identify commonalities between languages and better understand the nuances of language translation.
4. **Cross-cultural collaboration:** facilitate collaboration between individuals from different cultures and linguistic backgrounds. By identifying commonalities between languages and cultures, these tools could help to bridge communication gaps and foster more effective collaboration.

References

1. Vamvas, J. and Sennrich, R., 2022. NMTScore: A Multilingual Analysis of Translation-based Text Similarity Measures. arXiv preprint arXiv:2204.13692.
2. Muneer, Iqra, and Rao Muhammad Adeel Nawab. "Cross-Lingual Text Reuse Detection at sentence level for English–Urdu language pair." Computer Speech & Language 75 (2022): 101381.
3. Jain, A., Arora, A., Morato, J., Yadav, D. and Kumar, K.V., 2022. Automatic text summarization for Hindi using real coded genetic algorithm. Applied Sciences, 12(13), p.6584.
4. Gupta, H. and Patel, M., 2021, March. Method Of Text Summarization Using Lsa And Sentence Based Topic Modelling With Bert. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) (pp. 511-517). IEEE.
5. Zhang, P., Huang, X., Wang, Y., Jiang, C., He, S. and Wang, H., 2021. Semantic similarity computing model based on multi model fine-grained nonlinear fusion. IEEE Access, 9, pp.8433-8443.
6. Agarwala, S., Anagawadi, A. and Guddeti, R.M.R., 2021. Detecting Semantic Similarity Of Documents Using Natural Language Processing. Procedia Computer Science, 189, pp.128-135
7. Savytska, L.V., Vnukova, N.M., Bezugla, I.V., Pyvovarov, V. and Sübay, M.T., 2021. Using Word2vec technique to determine semantic and morphologic similarity in embedded words of the Ukrainian language.
8. Tiyajamorn, N., Kajiwara, T., Arase, Y. and Onizuka, M., 2021, November. Language-agnostic Representation from Multilingual Sentence Encoders for Cross-lingual Similarity Estimation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 7764-7774).
9. Ham, J. and Kim, E.S., 2021, November. Semantic alignment with calibrated similarity for multilingual sentence embedding. In Findings of the Association for Computational Linguistics: EMNLP 2021 (pp. 1781-1791).
10. Jeyaraj, M.N. and Kasthurirathna, D., 2021. MNet-Sim: A Multi-layered Semantic Similarity Network to Evaluate Sentence Similarity. arXiv preprint arXiv:2111.05412
11. Roostaee, M., Sadreddini, M.H. and Fakhrahmad, S.M., 2020. An effective approach to candidate retrieval for cross-language plagiarism detection: A fusion of conceptual and keyword-based schemes. Information Processing & Management, 57(2), p.102150.
12. Wang, X., Dong, X. and Chen, S., 2020, June. Text duplicated-checking algorithm implementation based on natural language semantic analysis. In 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC) (pp. 732-735). IEEE.
13. Jatnika, D., Bijaksana, M.A. and Suryani, A.A., 2019. Word2vec model analysis for semantic similarities in english words. Procedia Computer Science, 157, pp.160-167.
14. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q. and Artzi, Y., 2019. Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.
15. Xia, C., He, T., Li, W., Qin, Z. and Zou, Z., 2019, July. Similarity analysis of law documents based on Word2vec. In 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C) (pp. 354-357). IEEE.

16. Sunilkumar, P. and Shaji, A.P., 2019, December. A survey on semantic similarity. In 2019 International Conference on Advances in Computing, Communication and Control (ICAC3) (pp. 1-8). IEEE.
17. Haneef, I., Adeel Nawab, R.M., Munir, E.U. and Bajwa, I.S., 2019. Design and development of a large cross-lingual plagiarism corpus for Urdu-English language pair. Scientific Programming, 2019.
18. Bakhteev, O., Ogaltsov, A., Khazov, A., Safin, K. and Kuznetsova, R., 2019, September. CrossLang: the system of cross-lingual plagiarism detection. In Workshop on Document Intelligence at NeurIPS 2019.
19. Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
20. Som, S., 2019. Analysis of Natural Language Processing (NLP) approaches to determine semantic similarity between texts in domain-specific context (Master's thesis).
21. Shajalal, M. and Aono, M., 2018, September. Semantic textual similarity in bengali text. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP) (pp. 1-5). IEEE.
22. Pontes, E.L., Huet, S., Linhares, A.C. and Torres-Moreno, J.M., 2018. Predicting the semantic textual similarity with siamese CNN and LSTM. arXiv preprint arXiv:1810.10641.
23. Cherroun, H. and Alshehri, A., 2018, April. Disguised plagiarism detection in Arabic text documents. In 2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP) (pp. 1-6). IEEE.
24. Şenel, L.K., Utlu, I., Yücesoy, V., Koç, A. and Çukur, T., 2018, December. Generating semantic similarity atlas for natural languages. In 2018 IEEE spoken language technology workshop (SLT) (pp. 795-799). IEEE.
25. Xylogiannopoulos, K., Karampelas, P. and Alhajj, R., 2018, August. Text mining for plagiarism detection: multivariate pattern detection for recognition of text similarities. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 938-945). IEEE.
26. Kherwa, P. and Bansal, P., 2017, September. Latent semantic analysis: an approach to understand semantic of text. In 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC) (pp. 870-874). IEEE.
27. Shenoy, N. and Potey, M.A., 2016. Semantic similarity search model for obfuscated plagiarism detection in Marathi language using Fuzzy and Naïve Bayes approaches IOSR. Journal of Computer Engineering, 18(3), pp.83-88
28. Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Acknowledgement

We are highly indebted to our project guide **Dr. Madhu Nashipudimath** for their guidance and constant supervision as well as for providing necessary information regarding the project & also helped us in applying the knowledge that we have acquired during the semester and learning new concepts.

We would like to express our special thanks to our project coordinator, **Dr. Sheetal Gawhande**, for giving us the opportunity to do this project.

We would also like to thank the Head Of Department Of Information Technology **Dr. Satishkumar Varma** Sir for giving us an opportunity to understand, learn and implement this project, which helped us a lot.

We thank our Principal **Dr. Sandeep Joshi** for Providing us with all the facilities and required equipment for the project.

Durgesh Bhadane

Janhavi Jadhav

Sharayu Mane

Vandana Pabale