

A CNN-Based Hybrid Approach for Semantic Textual Similarity in Kannada–English Language Pair

¹Megha V, ²Pooja M R

¹Vidyavardhaka College of Engineering, Mysuru, India

megha.v@vvce.ac.in

²Vidyavardhaka College of Engineering, Mysuru, India

pooja.mr@vvce.ac.in

HOW TO CITE:

Megha V, Pooja M R (2026). A CNN-Based Hybrid Approach for Semantic Textual Similarity in Kannada–English Language Pair. International Journal of Special Education, 41(4s), 689-708.

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors.
Open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

ABSTRACT:

Semantic similarity estimation across a pair sentence is one among the major concerns for NLP, and Information retrieval, machine translation, text summarization, and automatic evaluation are just a few of its many uses. While much work has been done in monolingual settings, (STS) cross-lingual semantic textual similarity remains an open issue, particularly when considering low-resource language pairs such as Kannada-English. This research proposes a semantic textual similarity assessment method using a hybrid technique of a brief English and Kannada text sentence pairs by considering lexical, syntactic, and semantic aspects. The approach incorporates word-based similarity measures along with lexical resources and vector-based similarity measures along with word embeddings. Cross-lingual word embedding alignment is done by utilizing VecMap and MUSE. A lexical decomposition approach is used to decompose similar and dissimilar semantic components. Similarity in word order is incorporated to record syntactic structure. Finally, a scoring-level fusion method is employed to generate similarity score. Moreover, CNN model is also applied for better classification results. Efficiency of the model is evaluated based on Samanantar Parallel Corpus, which contains sentence pairs in different languages generated through AI4Bharat. The hybrid method outperforms the standard word-level and embedding-level techniques, as shown in the experimental results. Furthermore, It has been noted that the embedding alignment approach performs better as the size of the bilingual dictionaries, and MUSE outperforms other techniques for large dictionaries. The proposed framework is effective for cross-lingual semantic similarity measurement in low-resource scenarios and can be extended to other multilingual tasks pertaining to NLP such as automatic grading, translation systems in machines, and cross-language IR systems.

Keywords: Natural Language Processing (NLP), Semantic Textual Similarity (STS), Word Embeddings, Stop-word removal ,Part-of-Speech Tagging ,Word Sense Disambiguation (WSD).

1. Introduction

Most prominent and difficult challenges in the field of NLP is figuring out how similar two texts are semantically. A substantial amount of textual data has been generated and made digitally accessible in recent years through websites, online news articles, social media, etc. The rapid growth of textual data has created a pressing need for development of efficient methods for determining semantic similarity of two text passages within a set of documents. Many NLP tasks, including summarization of text, IR systems, categorization of text, and other systems like Q&A i.e question answering and machine translation systems, and web search engines, heavily rely on computation of semantic similarity. Computation of semantic similarity helps one to identify whether the pair of text convey equivalent meaning even if the words and sentences used in both pieces of text are completely different. The earlier methods for computing similarity between two pieces of text relied on statistical methods such as TF-IDF and BoW model for representing the textual data. Statistical methods rely on lexical features for determining how similar two texts are to one another. However, it is also found that these approaches cannot understand the underlying semantics of the text. The first limitation is that words have multiple meanings based on their usage, and different terms might be used to convey the same notion.

For example, in the below considered two sentences:

“Tony and Peter studied math and physics.”

“Tony studied math and Peter studied physics.”

In the above sentences, although words used are the same, the meaning of the two sentences is completely different. In the same way, the meaning of two other sentences, “John loves athletics” and “John is fond of sports,” is

completely the same, despite the fact that the concept is expressed using different terms.

To get beyond the drawbacks of conventional methods, semantic-based approaches for measuring text similarity have also been proposed. Semantic Textual Similarity (STS), one among the most important concepts in semantic-based approaches for measuring text similarity, is a "measure of the degree to which two text segments have the same meaning." STS measures the two sentences' semantic link is based on conceptual equivalence rather than the words used to describe them and assigns a score or rank to each pair of sentences according to how close they are semantically.

Semantic Textual Similarity (STS), is most important concepts in semantic-based approaches to measuring text similarity, is a gauge of how closely two text passages have the same meaning. STS evaluates two phrases' semantic relationship based on conceptual equivalency rather than the words used to describe them and assigns a score or rank to each pair of sentences according to how close they are semantically.

In the present work, we outline an approach to determine cross-lingual semantic similarity among Kannada & English. Part of Southern Indian state of Karnataka, almost 60 million people speak Kannada. The AICTE has been translating textbooks and course materials into several Indian languages, including Kannada, in recent years. Cross-lingual semantic similarity systems have sizable potential market in India because of this program, which has greatly raised demand for multilingual educational technologies. India has all these official languages, over 22 I think, and they are pushing more for using regional ones in schools and tests. That makes sense because not everybody is comfortable with English all the time. There can be English questions too, but a lot of students write answers in their own language,

like Kannada or whatever. So, in this setup from Figure 1, there's a module that checks how similar an English answer key is to a student's answer in Kannada. It's for automatic grading in places where languages mix up like that. If it's digital, you can just plug it in somewhere to help teachers decide grades or something.

This semantic similarity appears to be useful in determining whether the semantic meaning is the same, even across languages. This could be helpful in creating an automated system to grade, and perhaps even verify, the output of automated translation programs. While I am not entirely sure how this does everything perfectly, At the very least, I believe this might save a lot

of labour in multilingual teaching. The following is a list of contributions this research work focusing towards, according to the document:

- A novel approach to compute STS across Kannada and English phrases, using the techniques of convolutional neural networks, embedding space alignment, and lexical decomposition. We believe that this is the initial effort to tackle, the issue to our benefit of our knowledge and understanding, STS for a pair of Kannada and English languages.
- Accuracy and correlation of suggested approach are assessed in addition to the study of alignment at the embedding space at the word level.

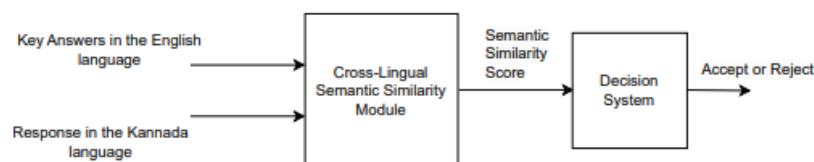


Figure 1. The calculation of cross-lingual semantic similarity in multilingual response paper evaluation is depicted [77].

2. Related Work

Individuals have thought of various ways to determine the similarities in meaning among sentences. Most of these are based on the research done in [5-10]. In these research papers, they make use of things like words, sentence structures, and the deeper meanings of the sentences to compare the similarities in meaning among sentences. I think these are the most basic building blocks. In the research paper [11], they discuss top three main entities involved in calculation of semantic similarities among sentences. Word embedding is one which maps the words in sentences to a vector space, which in turn represents similarities among the words in the sentences. Word weighting is another, which determines the importance of words in sentences. And then there are similar methods, which determine the similarities among the words in sentences, using word embedding concept. Clustering algorithms are one of the most used algorithms in the calculation of the similarities among words or sentences. Sometimes, I think the word

embedding concept is the most appropriate in the calculation of the similarities among sentences, as it considers the complexities in the sentences.

In the research paper [12], they focus on the Arabic language words, in which they determine the similarities among the words in the sentences using the lexical features, syntax, and semantics of the words in the sentences. After that, they employ ML algorithm, which will map features to the score using the support vector machine and linear regression algorithms.

Uncertainty is frequent problem when dealing with text, and [13] proposes something better with probabilistic rough set theory. It uses a mix of lower approximation similarities and upper ones to determine similarities for phrases. This might help with fuzzy things in the data. It is not all clear and simple, though. One thing that really stands out is the way these models deal with uncertainty in text data. Like, with the use of probabilistic tolerance relations, it helps manage the ambiguity that is always present in

text and sentences. It is kind of effective for that. Then there is this approach that these authors propose in reference [14]. They came up with a method for calculating similarities for sentences, and it is based on something called the ACVT kernel, which is a tree kernel method. It uses things like syntactic and semantic features, and this weighting thing for deep relationships.

I think that makes sense for capturing more than just surface level. Fuzzy Rough Set Theory comes up a lot too. It's been used to compute sentence similarity by mixing fuzzy logic with rough set ideas. This way, it handles vagueness and uncertainty in the text better, and that improves how accurate the semantic similarity estimates are. The theory itself is common for dealing with imprecision and incompleteness in textual data, as mentioned in [15]. Combining those elements lets you model ambiguous language more effectively, I guess. In studies [16] and [17], they went with word embeddings for text similarity. Sentence structure gets estimated through graph representations. They combine multiple similarity measures to make the computation more accurate. Its interesting how graphs can represent that structure. Similar idea in [18], where researchers proposed estimating semantic similarity for short texts. It uses word embeddings and some external sources. Each short paragraph turns into two dense vectors that capture the semantic content. That part feels a bit straightforward. Another study, [19], has this model based on three metrics. Optimal Alignment Similarity, or OptAlign, matches equivalent terms across texts using optimal word alignment. Greedy Association Similarity, GrAssoc, finds words matching with multilingual embeddings and a greedy approach. Then Aggregation Similarity, Aggreg, combines various metrics into one overall score. I'm not totally sure how well they all work together, but it seems like an attempt to cover different angles. This part gets a bit messy when you think about applying it. In addition, the authors proposed an embedding-based technique for the identification of latent subjects in short texts in the work presented in [20]. This

technique first identifies longer pseudo-texts by aggregating short texts and then identifies latent subjects in the pseudo-texts by using a Markov Random Field. Recently, deep learning techniques is been observed to perform better among several NLP tasks [21-24]. In the work presented in [25], Sentences' semantic similarity in medical literature is assessed by using a method based on the CNN. In a similar context, Siamese Neural Networks also employ RNN to assess similarity of phrases in pair of sentences [26, 27]. In the work presented in [28], the RCNN is proposed for assessment of semantic similarity of sentences. Here in proposed technique, the Siamese multilayer CNN is employed to assess the importance of the features of the sentences. The final score is obtained by fusing various sentence representations through the fusion layer.

In another research work, the authors proposed a three-layer structure comprising: lexical, syntactic, and semantic to enhance accuracy of sentence similarity calculation. Adding to it another model based on attention, comprising word n-grams generated using a bidirectional RNN, was proposed in [30] for determining how similar the provided sentences are semantically. Another algorithm was introduced in [31] for computing similarity by using similar as well as dissimilar aspects of the words. In another research work, a different approach was proposed in [32] for evaluating similarity across the given sentences using the concept of **Semantic Textual Similarity upon performing line-by-line alignment. The alignment score was calculated through the application of Multiple Linear Regression, while the classification techniques such as K-Nearest, SVM, and GNB i.e, Gaussian Naïve Bayes were used for multi-classification purposes. In yet another research paper, the authors suggested the application of following to evaluate semantic similarity of phrases in Bengali, there are almost six statistical measures used, which include frequency, matching ratio, parts of speech tag, sense matching, cosine similarity, & entropy. In yet another research work, a hybrid technique was suggested in [34], where a fine-tuned model of

the BERT algorithm is applied with weights to extract features. Deep model built on a bidirectional interaction network, which records semantic interactions between sentences To achieve further advances, a deep neural network, a deep fusion layer, and an attention mechanism were combined. To evaluate semantic similarity, another approach based on gated networks was proposed in [36]. The approach combines the dispersed representations with one-hot encoding. The authors of another investigation, [37], proposed the W-KG2Vec model, which is based upon the use of meta-path. Semantic linkages and text similarity between phrases are both used in the model. Another model, BERT-Text2Vec, combines pre-trained BERT model with LSTM encoder. Adding on to it sentiment lexicons and semantic models have been combined with word representations for the detection of semantic similarity between texts, as proposed in another investigation [38]. Another approach combines the use of knowledge bases and corpora for the evaluation of text similarity. The authors of another investigation [40] introduced the KNetwork, which is a unique sentiment analysis framework, especially for texts in different language. The phrase sentiment and phrase length are two basic elements of prior knowledge that have been proposed by the authors of [41] to be applied in a unique manner to incorporate many aspects to enhance classification performance.

As an effective learning of cross-lingual representations, pre-trained cross-lingual model known as XLM-RoBERTa is used in the investigation. The study also emphasizes how important it is to combine coarse-grained and fine-grained data for efficient sentence similarity computation. To address the lack of labelled datasets for machine learning model training, the authors of another study [42] suggested a technique for Sanskrit texts sentiment analysis . Suggested approach classifies Sanskrit texts sentiment analysis using zero-shot cross-lingual mapping without the need for direct sentiment annotation. To improve the interpretability of deep neural models for NLP tasks, authors of another study [43] suggested a framework for

describing the behaviour of LLMs by examining communication patterns of neurons in suggested architecture. In another research paper [44], the authors proposed a technique for rumour and GANs are made used to detect bogus news which means Generative Adversarial Networks. The proposed architecture has two components: a generative component for generating synthetic data and a discriminative component for differentiating real and fake information. In another research paper [45], During the COVID-19 pandemic, the authors suggested a network approach for identifying false information. Numerous techniques for calculating sentence similarity have been established, and numerous new techniques are documented in research articles [46–53], according to a survey of the literature. However, only a few research papers have been focused on computing semantic similarity across Kannada and English phrases. Considering the mentioned research gap in mind, this research aims at developing an effective semantic similarity computing framework across Kannada and English phrases. The method used to accomplish objective is explained in detail in Section 3 of this paper.

Although many methods have been developed for computing STS across sentences, most of the research papers are focused on computing STS between sentences of highly resource-rich languages like English and Chinese. Many methods are based on lexical and statistical measures, including TF-IDF and the Bag-of-Words, which are based upon shallow-level features and do not consider semantic features between sentences. Many recent methods, including BERT, XLM-RoBERTa, and CNN's, have been beneficial in computing semantic similarity between sentences. However, most of the recent methods have been tested on monolingual data or on highly resource-rich language pairs, less research being done on how well they perform in resource-constrained languages.

Therefore, performance of llanguages with limited resources has been relatively Insufficiently researched. In broader context of

Indian languages and Kannada, there has been limited annotated data and computational resources for semantic similarity tasks. In addition, there has been limited investigation of cross-lingual semantic similarity across Kannada and English, with a focus on short text sentences. Existing approaches for this task are mostly based on translation-based approaches or simple embeddings. Such approaches are likely to be inadequate for capturing semantic relationships between Kannada and English sentences, considering the structural and linguistic differences between the two languages. Another limitation of existing approaches has been the insufficient integration of lexical features, semantic embeddings, and deep learning models into a single framework for cross-lingual semantic similarity computation. In addition, there has been limited investigation of embedding alignment and feature extraction using deep learning models for Kannada-English sentence pairs. Thus, by combining lexical characteristics, embeddings, and deep learning models, it is necessary to create an effective framework for cross-lingual semantic similarity that can model semantic links between Kannada and English phrases. The above approach seems to have potential application in improving the efficiency of multilingual NLP applications like answer scoring, translation assessment, and cross-lingual information retrieval in a multilingual nation such as India.

3. Methodology

The existing Semantic Textual Similarity (STS) techniques for short texts generally utilize the following techniques to estimate the similarity across sentence pairs: word-based methods or vector-based methods. Lexical database is utilized in the word-based strategy to ascertain the semantic similarity across word pairs in the phrases. Outcome is obtained by assessing both the word alignment in between the sentences and the semantic similarity across the word pairs. Word vector representations are produced using the DNN in the vector-based technique. In vector-based approaches, In continuous vector space, a word is expressed as a vector in numerical form. In this approach, huge amounts

of text data are utilized during the training process for learning vector representations. Vector representation can carry the semantic relationship, context dependency, as well as syntactic information among the words. The vector-based approach can achieve a more comprehensive representation in contrast to traditional lexical approach. However, word-based approach is also capable of achieving the desired result through the effective utilization of the lexical database. Therefore, blending of vector-based approach with word-based approach can achieve desired result.

In the presented study, we propose hybrid method which considers both the word-based & vector-based approaches for evaluating the semantic similarity of short phrases within the scope of Kannada-English sentence pairs. Hybrid synthesis approach measures the similarity through both semantic and lexical methods. The detailed explanation regarding word-based & vector-based approaches is provided in the sections that follow.

3.1 Comparing Vector-Based with Word-Based Similarity

A lexicon database like WordNet could help calculate semantic similarity between words [54–56]. WordNet is a lexical database containing a hierarchy among its entries. In WordNet, words are clustered into groups referred to as synsets based on their cognitive synonyms. Synsets are unique semantic representations of words that are hierarchical in nature for various speech components. Most of similarity computation methodologies using words tend to use an almost identical algorithm. Initially, POS tagging is carried out on both the sentences, i.e., SSS and TTT. Following POS tagging, the word sense of the individual words needs to be found out using Word Sense Disambiguation. This will make it possible to understand the word's proper meaning when it appears in varied circumstances. For instance, the word "bank" can refer to both a financial organization and the riverbank. Knowing a word's meaning will enable one to determine its semantic relatedness. In this case, WordNet will be used for

calculating the semantic similarity between pairs of noun-noun and verb-verb words $s_i \in S_{s_i} \setminus \text{in } S_{s_i} \in S$ and $t_j \in T_{t_j} \setminus \text{in } T_{t_j} \in T$. The semantic similarity is measured based on the shortest path length between the two words in hierarchy tree[57]. These methods help identify the best word alignment across the two sentences. One of the advantages associated with lexical database similarity methods is that they offer a mechanism to handle unknown terms in the vocabulary by leveraging semantic relations in the knowledge base. In contrast, vector-based methods represent a sentence SSS as a vector of words $[s_1, s_2, \dots, s_m]$ $[s_1, s_2, \dots, s_m]$ $[s_1, s_2, \dots, s_m]$, where each word s_i is represented using a pre-trained **Word Embedding**. These embeddings are generated using deep learning models trained on large corpora. In such methods, the similarity between two short sentences SSS and TTT is typically computed by evaluating the similarity across word pairs (s_i, t_j) (s_i, t_j) (s_i, t_j) and determining which matching pairs work best. But these approaches may not preserve word order, and there are possibilities of handling **OOV (out-of-vocabulary)** words, particularly in applications that are domain-based. For example, in **NLP** applications in healthcare, words related to symptoms, diseases, or medical conditions may appear in different word order while carrying different meanings. When describing a sentence's syntactic structure, word order is

important. As examples, let's look at two sentences:

- “Your dog chased my cat.”
- “My cat chased your dog.”

Though both sentences use the same words, they carry different meanings due to word order differences. The syntactic structures, represented using parse trees, are different in both sentences. Despite the fact that the English language follows a word order pattern strictly, there is some flexibility in word order at the phrase level [58]. Hence, word order preservation is significant while computing similarity scores for sentences. Figure 2 shows the suggested architecture for computing semantic similarity, while the detailed architecture is presented in Figure 3. In this framework, sentences SSS and TTT are in two different languages. In this research, semantic similarity computation is carried out between **Kannada** and English short sentences.

In preprocessing step, several operations are carried out, which include tokenization, removal of stop words, and **WSD**. Although word embeddings can capture contextual information to some extent, applying WSD to English words further enhances semantic interpretation by identifying each word's most suitable meaning in the context. The overall accuracy of the semantic similarity calculation is increased by this step.

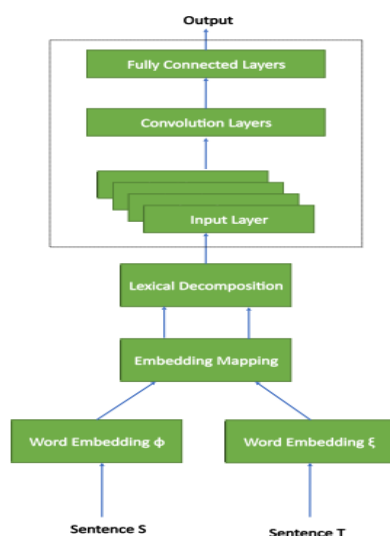


Figure 2: A suggested architecture for computing semantic similarity [77].

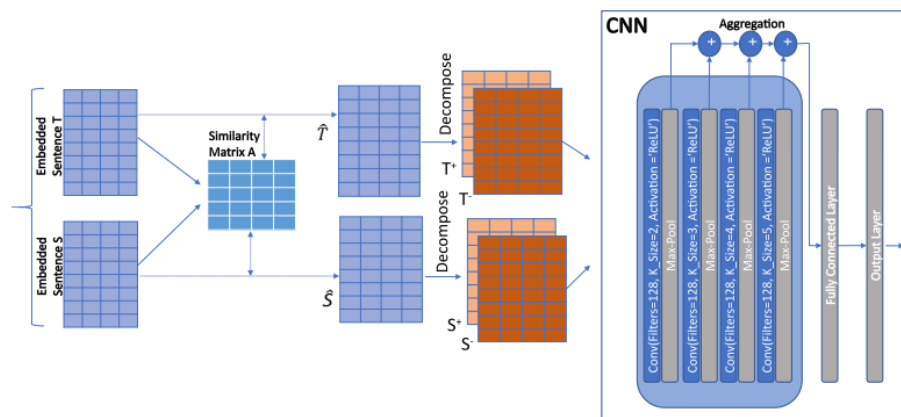


Figure 3 shows a comprehensive cross-lingual semantic similarity calculation architecture [77].

3.2 Word Embedding

In order to transform textual data into numerical form, we employ the Word Embedding method, whereby each word found in every sentence is transformed into dense vector form. Here, every word $s_i \in S_i$ or $t_j \in T_j$, expressed in a vector form of dimensionality d , is derived by a pre-trained neural network model [60]. The idea behind word embedding is to map words into continuous vectors such that words with similar semantic meanings will be closer together.

Out of all the embedding models we have Word2Vec, GloVe, and FastText are widely used. Word2Vec algorithm makes use of Skip-Gram model or CBOW model to create embeddings. Here, n -word sentences are converted into embeddings of the form $n \times d$, where d stands for the embedding dimensionality.

Embedding process, converts each single word into its equivalent corresponding vector representation in d -dimensional space. For a sentence $T = [t_1, t_2, \dots, t_n]$, $T = [t_1, t_2, \dots, t_n]$, and for another sentence $S = [s_1, s_2, \dots, s_m]$, $S = [s_1, s_2, \dots, s_m]$ [77]. As a result, both sentences are transformed into matrices of word embeddings, enabling similar computation between their semantic representations.

Pre-trained embedding models fall into two groups:

1. Monolingual Embeddings: Text from a

single language is used to train these models.

2. Cross-Lingual Embeddings: These models, which occasionally include word alignment data, are trained on parallel corpora comprising several languages [61].

This study investigates both monolingual and cross-lingual embeddings for phrase representation. Pre-trained models are used to integrate the processed phrases into 300-dimensional vectors. The spaCy library's tok2vec model is used to incorporate English texts [62]. FastText [63], which uses Skip-Gram architecture by default options explained in [64], is also used to incorporate Kannada text. The FastText repository makes pre-trained Kannada word embeddings accessible to the general population.

A major strength of Fast Text lies in its implementation of sub-word embeddings using character n -grams. This feature enables Fast Text to handle morphologically rich languages like Kannada. Furthermore, it mitigates the problem of out-of-vocabulary words through the construction of embeddings using character-level information.

3.3 Monolingual Embedding and Alignment

Since separate embeddings are created for English and Kannada language, alignment between these two embedding spaces is needed, where words having the same meaning in the two languages will be represented using the same vectors. For creating alignment between

the embeddings, we have employed techniques like MUSE and VecMap [65–69].

MUSE and VecMap offer support for supervised and unsupervised alignment methods:

- **Supervised Alignment:** This method relies on a bilingual dictionary to train a mapping between the embeddings of two languages.
- **Unsupervised Alignment:** This method relies on automatic training for the mapping between the embeddings of two languages without a dictionary.

One such impediment for achieving aligned corpora is the absence of a bilingual lexicon in the language pair Kannada-English. There is only one bilingual lexicon in Kannada-English which consists of 4,000 words approximately. Hence, an alternative alignment technique is considered to achieve better results.

By aligning embedding space of the Kannada and English languages, words semantically similar to each other from it is possible to map both languages onto a common area. This will help in the computation of cross-lingual semantic similarity between sentence pairs.

3.4 Lexical Decomposition and Similarity Calculation

To further improve the semantic similarity evaluation between sentence pairs, we apply a **lexical decomposition technique** which separates semantic components of words into **similar** and **dissimilar** parts. This method helps capture fine-grained semantic relationships between words across sentences.

Two phrases $T=[t_1, t_2, \dots, t_n]$ and $S=[s_1, s_2, \dots, s_m]$ are converted into vector representation using Word Embedding methods. After obtaining embeddings, semantic similarity across words from the two sentences is calculated.

3.4.1 Word Similarity Matrix

To quantify semantic relationship across each pair of words from two phrases, a similarity matrix MMM is created. Cosine Similarity is made use of to calculate each matrix element:

$$M_{i,j} = \cos(\theta_{i,j}) = \frac{s_i \cdot t_j}{\|s_i\| \|t_j\|} = \frac{s_i \cdot t_j}{\|s_i\| \|t_j\|}$$

where

- s_i is the embedding vector of the i th word in sentence SSS,
- t_j is the embedding vector of the j th word in sentence TTT.

Similarity matrix captures the semantic relationship between all word pairs.

3.4.2 Lexical Decomposition

For each word vector, the semantic representation is decomposed into two components:

1. **Similar Component** – represents the semantic information shared with the corresponding word in the other sentence.
2. **Dissimilar Component** – represents the semantic information that is not shared between the two words.

For a word s_i , its similar component s_i^+ and dissimilar component s_i^- are computed as: $s_i^+ = \alpha_i s_i$ and $s_i^- = (1 - \alpha_i) s_i$ where α_i represents the similarity weight derived from the similarity matrix. Similarly, the components t_j^+ and t_j^- are computed for words in sentence TTT. This decomposition enables the model to separate **shared semantic information** from **non-overlapping information**, which improves similarity evaluation.

3.4.3 Sentence-Level Similarity

The process of decomposition is followed by the computation of the similarity in between the two phrases, where the similarity across the word pairs is combined to generate the similarity

between the two sentences. The similarity score is represented as:

$$\text{Sim}(S,T) = \frac{\sum_{i=1}^m \sum_{j=1}^n M_{i,j} \times \text{Sim}(S,T)}{M_{i,j} \times m \times n}$$

where m and n represent number of words in sentences SSS & TTT, respectively.

a. Word-Order Similarity

Although measures of semantic similarity try to capture the meanings of sentences conceptually, it is equally important to take into consideration the ordering of the words in a sentence. The sequence of the words is very critical in determining the syntax of the sentence. Two different sentences can have similar words, but by changing their order, they would mean totally different things. For example:

- “Your dog chased my cat.”
- “My cat chased your dog.”

Though both sentences contain similar words, their meanings differ considerably because of the change in word order. Therefore, incorporating word-order information helps improve the accuracy of **Semantic Textual Similarity** computation.

3.5.1 Word-Order Vector Representation

Word order can be captured by creating a word order vector of sentences. If the sentences are given as follows,

$T = [t_1, t_2, \dots, t_n]$ $T' = [t'_1, t'_2, \dots, t'_n]$
 $S = [s_1, s_2, \dots, s_m]$ $S' = [s'_1, s'_2, \dots, s'_m]$. Then the union vocabulary set is formed using these sentences. Every word from this union set is indexed in terms of the positions they occupy in sentences. Word order vectors are made using these sentences and indices, and each word of these vectors represents the position of the t -word in sentences. For example:

Sentence SSS:
 “John loves sports”

Sentence TTT:
 “John likes athletics”

Combined vocabulary:

{John, loves, sports, likes, athletics}

Word-order vectors:

$$\text{OS} = [1, 2, 3, 0, 0] \quad \text{O}_S = [1, 2, 3, 0, 0] \quad \text{OS} = [1, 2, 3, 0, 0] \quad \text{OT} = [1, 0, 0, 2, 3] \quad \text{O}_T = [1, 0, 0, 2, 3] \quad \text{OT} = [1, 0, 0, 2, 3]$$

Here, **0** indicates that the word does not appear in that sentence.

3.5.2 Word-Order Similarity Calculation

The similarity between the two word-order vectors is calculated using the following formula:

$$\text{Sim}_{\text{order}}(S,T) = 1 - \frac{\|\text{OS} - \text{OT}\|}{\|\text{OS} + \text{OT}\|}$$

where:

- OS_S = word-order vector of sentence SSS
- OT_T = word-order vector of sentence TTT
- $\|\cdot\|$ represents the Euclidean norm.

This measure evaluates how similar the word positions are between the two sentences.

3.5.3 Final Similarity Score

The overall sentence similarity is obtained by combining:

1. **Semantic similarity** (from lexical and embedding methods)
2. **Word-order similarity**

$$\text{Sim}_{\text{final}} = \alpha \times \text{Sim}_{\text{semantic}} + (1 - \alpha) \times \text{Sim}_{\text{order}}$$

where:

- $\text{Sim}_{\text{semantic}}$ represents the semantic similarity score

- $\text{SimorderSim}_{\{\text{order}\}}\text{Simorder}$ represents the word-order similarity score
- α is a weighting parameter between 0 and 1.

3.5.4 Importance of Word-Order Similarity

Considering word-order similarity provides several advantages:

- Captures syntactic structure of sentences
- Distinguishes sentences with identical words but different meanings
- Improves sentence similarity estimation in short texts
- Improves performance when combined with Deep Neural Networks models.

In the proposed framework, word order similarity is combined with lexical similarity and embedding-based similarity to yield final semantic similarity across the Kannada and English sentence pairs.

4. Score-Level Fusion

To obtain semantic similarity across two sentences, the scores obtained from different modules of proposed framework are combined using the **score level fusion** method. Using this method, the individual similarities obtained using different methods can be combined to obtain the final similarity score.

In computation of sentence similarity, different methods can capture different aspects of textual information. For example, in the computation of lexical similarity, the relationship between words is made use. Moreover, in the calculation of word embedding-based similarity, contextual relationship between words is used. Furthermore, the word order-based similarity captures the syntactic relationship between words. Since different methods capture different aspects of textual information, the combination of individual similarities can capture the overall textual relationship.

Let the individual similarities obtained using different methods be represented as:

- $\text{SlexS}_{\{\text{lex}\}}\text{Slex}$ – lexical similarity score
- $\text{SvecS}_{\{\text{vec}\}}\text{Svec}$ – The similarity score based on vectors
- $\text{SorderS}_{\{\text{order}\}}\text{Sorder}$ – The similarity score based on word-order The final similarity score is computed using a weighted fusion strategy:

$$S_{\text{final}} = \alpha \cdot \text{Slex} + \beta \cdot \text{Svec} + \gamma \cdot \text{Sorder}$$

where:

- α, β, γ are the weighting parameters assigned to the individual components of the similarities
- $\alpha + \beta + \gamma = 1$

These parameters control the level of importance assigned to individual similarities during the computation of the final level of similarity. The optimal level of these parameters is determined through experimentation to attain optimal performance. It is to be noted that the advantages of using the score level fusion include the following:

Improved accuracy is obtained through the fusion of complementary information with different perspectives, such as lexical, semantic, and syntactic. It eliminates the limitations imposed by individual approaches. Robustness is improved during cross-lingual scenarios with linguistic variations. In the proposed system, the score level fusion is employed after the computation of the individual similarities between the Kannada and English sentence pairs. The resultant fusion is the Semantic Textual Similarity.

5. Experimentation and Results

The proposed method was implemented by utilizing the TensorFlow library, which is a deep learning library. The implementation was carried out using a computer with an 11th Generation Intel Core i5 processor, running at a frequency

of 2.70 GHz, and accompanied by 16 GB RAM and an NVIDIA GeForce RTX 3050 GPU with 8 GB RAM.

5.1 Preprocessing

Various preprocessing techniques were applied to input sentences before similarity computation. These steps include:

- Stop-word removal
- **Part-of-Speech Tagging**
- **Word Sense Disambiguation (WSD)**
- Word filtering using **WordNet**

The above-mentioned preprocessing steps helps in removing irrelevant tokens, identify grammatical structures, and obtain contextual meanings of words, which in turn improves the accuracy of semantic similarity evaluation.

5.2 Dataset Description

For testing and validating the performance of the proposed method, experiments were conducted by utilizing the **Samanantar Parallel Corpus** dataset. The Samanantar corpus offers parallel sentence pairs for English and other Indic languages. The Samanantar corpus can be accessed via the **AI4Bharat** initiative, which offers English paired with **11 Indic languages**, including **Kannada**.

The dataset sources include:

- WAT 2021 MultiIndicMT shared task
- OPUS parallel corpus
- Various online web resources

There are over 49.7 million parallel sentences between English and 11 Indic languages in the dataset. Among them are:

- 12.4 million sentences were taken from pre-existing datasets.
- Web scraping was used to gather 37.4 million sentences.

Because the dataset is pooled from various sources, it depicts text from **diverse domains**,

making it versatile for evaluating cross-lingual semantic similarity models.

5.3 Annotation for Semantic Similarity

During the preparation of the dataset, sentence pairs were extracted from the corpus scraping operation based on the similarity score calculated by LaBSE (Language-agnostic BERT Sentence Embedding).

A portion of the dataset was annotated for Semantic Textual Similarity.

The annotation scheme for the extracted data consisted of categorization into three similarity levels:

1. **Definitely Accept** – Sentences are semantically equivalent.
2. **Marginal Accept** – Sentences share partial semantic similarity.
3. **Reject** – Sentences are semantically different.

A total of 9.566K phrase pairs between English and 11 Indic languages were annotated. There are 0.957K annotated sentences in the Kannada–English pair.

The dataset was divided as follows:

- **Training data:** 70%
- **Testing data:** 30%

5.4 Assessment of Embedding Alignment

We employed a bilingual lexicon with **human annotations** as ground reality to assess the quality of cross-lingual embedding alignment. The embedding alignment task has been framed as a **word retrieval task**, where semantically equivalent words in the target language have been retrieved for each word in the source language.

The similarity between the word embeddings has been calculated by applying **Cosine Similarity**.

The performance of the word retrieval task has been measured by **precision at rank K**, defined by:

- **P@1** – Precision considering the top retrieved word
- **P@5** – Precision considering the top 5 retrieved words
- **P@10** – Precision considering the top 10 retrieved words

Here, **P@K** represents the precision calculated using the top KKK retrieved candidates for each query word. The evaluation strategy has been based on the procedure outlined in [19].

5.5 CNN Training Parameters

The parameters used in the Convolutional Neural Network (CNN) layer in the suggested approach are experimentally determined. The final form of the suggested approach for the experiment is as follows:

Value of the parameters considered:

ReLU Activation Function

There are 100 epochs

Rate of Learning: 0.001

Epsilon(ϵ) $1 \times 10^{-71} \times 10^{-7} \times 1 \times 10^{-7}$

Cross-Entropy Loss Function

ReLU activation function is used in the approach because of its efficiency in training deep neural networks and the absence of the vanishing gradient problem.

Both **VecMap** and **MUSE** are popular tools used for the task of learning **Cross Lingual Word Embeddings**, but the approaches used by these tools are different. The performance characteristics of the tools also vary. As mentioned in the results provided in Tables 1

and 2, the task of determining the superior performance between the two approaches is difficult, as the performance is dependent on the requirements and constraints imposed on the task.

The experimental results demonstrate that **MUSE** has an advantage when working with the **Kannada-English** pair of languages, where the number of word pairs in a bilingual dictionary is more than 2,000, as shown in Figure 5.

However, there is a difference in their requirements during training and flexibility in usage. **VecMap** mostly uses a **bilingual dictionary or a parallel corpus** as a source of information in aligning the vectors in the embeddings. It is therefore best suited in cases where there is a possibility of having access to these resources.

On the contrary, **MUSE** has shown great potential in enabling the learning of cross-lingual embeddings based on information obtained from a monolingual corpus. As a result, it works best in situations where access to parallel corpora and bilingual dictionaries is restricted.

Hence it is safe to conclude that in cases where access to larger bilingual dictionaries is available, **MUSE** has shown better results in terms of vector alignment compared to **VecMap**. **VecMap** is therefore best suited in cases where access to reliable parallel information is available.

These tables show the performance in retrieving the embedding alignment using **VecMap** and **MUSE**, measured by Precision at K (**P@1**, **P@5**, **P@10**) based on Cosine Similarity.

Table 1:

Method Dictionary Size	P@1	P@5	P@10
VecMap 500	0.41	0.55	0.63
VecMap 1000	0.47	0.61	0.70
VecMap 1500	0.52	0.66	0.74
VecMap 2000	0.56	0.69	0.77

Method Dictionary Size P@1 P@5 P@10

VecMap	2500	0.58	0.72	0.80
--------	------	------	------	------

Table 2:

Method Dictionary Size P@1 P@5 P@10

MUSE	500	0.39	0.53	0.61
MUSE	1000	0.45	0.60	0.68
MUSE	1500	0.51	0.65	0.73
MUSE	2000	0.57	0.71	0.79
MUSE	2500	0.61	0.75	0.83

5.6 Results and Discussion

This section includes a detailed study of the alignment performance for the Kannada-English language pair as well as the experimental findings for the proposed cross-lingual semantic similarity model. Both the performance of the suggested Convolutional Neural Network-based semantic similarity model and the efficacy of the cross-lingual embedding alignment techniques are used to assess the performance.

5.6.1 Embedding Alignment Evaluation

The embedding alignment challenge was developed as a word translation retrieval task in order to assess the quality of cross-lingual word embeddings. The work is carried out by using the Cosine Similarity between the word embedding vectors to retrieve the semantically

related terms in the target language for the supplied word in the source language. Precision at **K**, or **P@1**, **P@5**, and **P@10**, is used to evaluate the performance.

The effectiveness of the widely used cross-lingual embedding alignment tools, such as VecMap and MUSE, is used to assess the efficacy of the cross-lingual embedding alignment techniques. Tables 1 and 2 show the retrieval task accuracy for various bilingual dictionary sizes.

The findings of the experiment show that the cross-lingual embedding alignment techniques work well for the Kannada-English language pair. The size of the bilingual dictionary utilized during the alignment process is found to have an impact on performance.

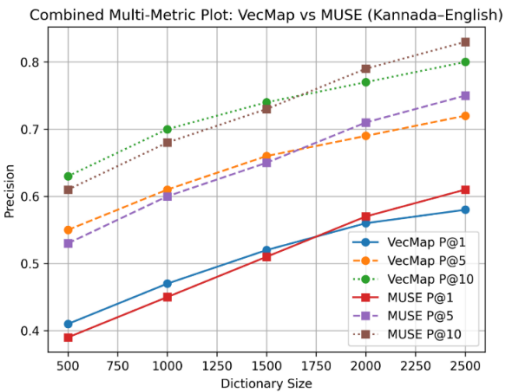


Figure 4: Embedding alignment retrieval performance using VecMap & MUSE [77]

5.7 Impact of Dictionary Size

The results show that both models exhibit a pattern of performance improvement as the size of the bilingual dictionary used in the model increases. This is demonstrated by the rise in P@1, P@5, and P@10 values, which show that there are more accurate translations in the result list.

5.8 Comparison Between VecMap and MUSE

Even though both VecMap and MUSE show competitive results, choosing a superior method is not a simple task. The availability of linguistic resources and the model's requirements for managing the word alignment problem determine which approach is best. From the experimental results, VecMap performs better than MUSE when a small size of a bilingual dictionary is used, as VecMap incorporates a series of supervised learning strategies with a high-quality seed dictionary. On the other hand, MUSE performs better than VecMap when a dictionary size larger than 2000 word pairs is used, as presented in Figure 5.

MUSE offers another advantage in learning cross-lingual embeddings using monolingual data, which is beneficial in scenarios where parallel data is not readily available in a language pair, a common situation in many low-resource language pairs.

5.9 Performance of the Proposed Model

The suggested semantic similarity model then employed the aligned embeddings as input characteristics. Annotated sentence pairs from the Samanantar Parallel Corpus—a collection of multilingual parallel data selected by AI4Bharat—were used to train the model. The model has successfully classified the sentence pairs into three different semantic similarity categories:

- Accept
- Marginal Accept
- Reject

The use of aligned cross-lingual embeddings has shown significant improvement in the model's capacity for semantic relationships between Kannada and English sentences. The representation of context-based semantics has been further improved by the application of deep neural networks, allowing the model to recognize the similarities and differences between the two sentences.

5.10 Discussion

The outcomes of the experiment have demonstrated that the accuracy of semantic similarity identification for multilingual sentences is much enhanced by the application of efficient cross-lingual embedding alignment. The findings have also demonstrated that using extensive multilingual resources can improve embedding alignment accuracy, which in turn improves semantic retrieval performance. Even though VecMap and MUSE have shown promising results, the use of these techniques is limited by the availability of resources for training. Moreover, the performance of these techniques is also limited by the nature of the target languages. The accuracy of cross-lingual semantic similarity detection has significantly improved with the usage of the suggested CNN-based semantic similarity model. Additionally, these techniques have shown a lot of promise for improving the performance of multilingual natural language processing. Additionally, these techniques have shown a lot of promise for improving the performance of multilingual natural language processing.

Conclusion

This study used deep learning-based classification models and cross-lingual embedding alignment to present the framework for cross-lingual semantic similarity among English and Kannada. Main aim behind the introduction of this model is to enhance the efficiency of Semantic Textual Similarity (STS) detection across two different language phrases.

To do this, the word translation retrieval task with the aid of Precision at K (P@1, P@5, P@10) utilizing the Cosine Similarity metric is

used to assess the effectiveness of the two widely used methods for cross-lingual word embedding alignment, VecMap and MUSE. It is evident from the experimental results that both methods are capable of successfully aligning the cross-lingual word embeddings.

But the performance depends on the size of the bilingual dictionary. VecMap can effectively align the word embeddings using the smaller dictionary. On the other hand, when the dictionary size exceeds 2000 word pairs, the MUSE model performs better. The CNN model is then used to assess the model's performance using annotated data sets created using the Samanantar Parallel Corpus, a multilingual parallel corpus created by the AI4Bharat

program. It is evident from the experimental findings that the suggested model is successfully classifying the sentence pairs into various semantic similarity classes. The experimental findings clearly demonstrate the effectiveness of the suggested model in improving the efficiency of the semantic similarity detection between two sentences in different languages. The effectiveness of the suggested model in improving the efficiency of the semantic similarity identification between two distinct language sentences is evident from the experimental findings. The experimental findings clearly demonstrate the effectiveness of the suggested model in improving the efficiency of the semantic similarity detection between two sentences in different languages.

References

- Pikuliak, M., Šimko, M., & Bieliková, M. (2021). Cross-lingual learning for text processing: A survey. *Expert Systems with Applications*, 165, 113765.
- Saad, M., Langlois, D., & Smaïli, K. (2014). Cross-lingual semantic similarity measure for comparable articles. In *Advances in Natural Language Processing* (pp. 105–115). Springer, Cham.
- Cer, D. M., Diab, M. T., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). SemEval-2017 Task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of SemEval@ACL*, Vancouver, Canada.
- Camacho-Collados, J., Pilehvar, M. T., Collier, N., & Navigli, R. (2017). SemEval-2017 Task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of SemEval@ACL*, Vancouver, Canada.
- Zhao, C., Wu, M., Yang, X., Zhang, W., Zhang, S., Wang, S., & Li, D. (2024). A systematic review of cross-lingual sentiment analysis: Tasks, strategies, and prospects. *ACM Computing Surveys*, 56, 1–37.
- Xu, Y., Cao, H., Du, W., & Wang, W. (2022). A survey of cross-lingual sentiment analysis: Methodologies, models and evaluations. *Data Science and Engineering*, 7, 279–299.
- Chandrasekaran, D., & Mago, V. (2021). Evolution of semantic similarity—A survey. *ACM Computing Surveys*, 54, 41.
- Prakoso, D. W., Abdi, A., & Amrit, C. (2021). Short text similarity measurement methods: A review. *Soft Computing*, 25, 1–25.
- Navigli, R., & Martelli, F. (2019). An overview of word and sense similarity. *Natural Language Engineering*, 25, 693–714.
- Khattak, F. K., Jeblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., & Rudzicz, F. (2019). A survey of word embeddings for clinical text. *Journal of Biomedical Informatics*, 100, 100057.
- Alian, M., & Awajan, A. (2020). Factors affecting sentence similarity and paraphrasing identification. *International Journal of Speech Technology*, 23, 851–859.
- Alian, M., & Awajan, A. (2021). Arabic sentence similarity based on similarity features and machine learning. *Soft Computing*, 25, 10089–10101.
- Yan, R., Qiu, D., & Jiang, H. (2021). Sentence similarity calculation based on probabilistic tolerance rough sets. *Mathematical Problems in Engineering*, 2021, 1–9.
- Quan, Z., Wang, Z. J., Le, Y., Yao, B., Li, K., & Yin, J. (2019). An efficient framework for sentence similarity modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27, 853–865.

- Chatterjee, N., & Yadav, N. (2019). Fuzzy rough set-based sentence similarity measure and its application to text summarization. *IETE Technical Review*, 36, 517–525.
- Kenter, T., & De Rijke, M. (2015). Short text similarity with word embeddings. In *Proceedings of the ACM International Conference on Information and Knowledge Management* (pp. 1411–1420).
- Farouk, M. (2020). Measuring text similarity based on structure and word embedding. *Cognitive Systems Research*, 63, 1–10.
- Nguyen, H. T., Duong, P. H., & Cambria, E. (2019). Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowledge-Based Systems*, 182, 104842.
- Glavaš, G., Franco-Salvador, M., Ponzetto, S. P., & Rosso, P. (2018). A resource-light method for cross-lingual semantic textual similarity. *Knowledge-Based Systems*, 143, 1–9.
- Qiang, J., Chen, P., Wang, T., & Wu, X. (2017). Topic modeling over short texts by incorporating word embeddings. In *Advances in Knowledge Discovery and Data Mining* (pp. 363–374). Springer, Cham.
- Otter, D. W., Medina, J. R., & Kalita, J. K. (2021). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32, 604–624.
- Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In *Proceedings of the Neural Information Processing Systems (NeurIPS)* (pp. 2042–2050).
- Hou, S. L., Huang, X. K., Fei, C. Q., Zhang, S. H., Li, Y. Y., Sun, Q. L., & Wang, C. Q. (2021). A survey of text summarization approaches based on deep learning. *Journal of Computer Science and Technology*, 36, 633–663.
- Alshemali, B., & Kalita, J. (2020). Improving the reliability of deep neural networks in NLP: A review. *Knowledge-Based Systems*, 191, 105210.
- Zheng, T., Gao, Y., Wang, F., Fan, C., Fu, X., Li, M., Zhang, Y., Zhang, S., & Ma, H. (2019). Detection of medical text semantic similarity based on convolutional neural network. *BMC Medical Informatics and Decision Making*, 19, 156.
- Chicco, D. (2021). Siamese neural networks: An overview. In *Artificial Neural Networks* (pp. 73–94). Humana Press.
- Ranasinghe, T., Orasan, C., & Mitkov, R. (2019). Semantic textual similarity with Siamese neural networks. In *Proceedings of RANLP* (pp. 1004–1011).
- Peng, S., Cui, H., Xie, N., Li, S., Zhang, J., & Li, X. (2020). Enhanced-RCNN: An efficient method for learning sentence similarity. In *Proceedings of The Web Conference* (pp. 2500–2506).
- Ferreira, R., Lins, R. D., Simske, S. J., Freitas, F., & Riss, M. (2016). Assessing sentence similarity through lexical, syntactic and semantic analysis. *Computer Speech & Language*, 39, 1–28.
- Lopez-Gazpio, I., Maritxalar, M., Lapata, M., & Agirre, E. (2019). Word n-gram attention models for sentence similarity and inference. *Expert Systems with Applications*, 132, 1–11.
- Wang, Z., Mi, H., & Ittycheriah, A. (2016). Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING*.
- Majumder, G., Pakray, P., Das, R., & Pinto, D. (2021). Interpretable semantic textual similarity using alignment and regression. *Applied Intelligence*, 51, 7322–7349.
- Das, A., Mandal, J., Danial, Z., Pal, A., & Saha, D. (2020). Bengali question answering using semantic similarity analysis. *International Journal of Speech Technology*, 23, 873–884.
- Viji, D., & Revathy, S. (2022). A hybrid approach of weighted fine-tuned BERT extraction with deep Siamese Bi-LSTM model for semantic text similarity identification. *Multimedia Tools and Applications*, 81, 1–27.
- Liu, M., Zhang, Y., Xu, J., & Chen, Y. (2021). Deep bi-directional interaction network for sentence matching. *Applied Intelligence*, 51, 4305–4329.

- Xiong, Y., Chen, S., Qin, H., Cao, H., Shen, Y., Wang, X., Chen, Q., Yan, J., & Tang, B. (2020). Distributed representation and one-hot representation fusion with gated network for clinical **Semantic Textual Similarity**. *BMC Medical Informatics and Decision Making*, 20, 72.
- Do, P., & Pham, P. (2021). W-KG2Vec: A weighted text-enhanced meta-path-based knowledge graph embedding for similarity search. *Neural Computing and Applications*, 33, 16533–16555.
- Araque, O., Zhu, G., & Iglesias, C. A. (2019). A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowledge-Based Systems*, 165, 346–359.
- Yang, J., Li, Y., Gao, C., & Zhang, Y. (2021). Measuring short-text similarity based on semantic and syntactic information. *Future Generation Computer Systems*, 114, 169–180.
- Jain, A., Jain, G., & Tewari, D. (2024). KNetwork: Advancing cross-lingual sentiment analysis for enhanced decision-making in linguistically diverse environments. *Knowledge and Information Systems*, 66, 1–19.
- Wang, L., Liu, S., Qiao, L., Sun, W., Sun, Q., & Cheng, H. (2022). A cross-lingual sentence similarity calculation method with multi-feature fusion. *IEEE Access*, 10, 30666–30675.
- Kumar, P., Pathania, K., & Raman, B. (2023). Zero-shot learning-based cross-lingual sentiment analysis for Sanskrit text with insufficient labeled data. *Applied Intelligence*, 53, 10096–10113.
- Xiao, X., Zhou, C., Ping, H., Cao, D., Li, Y., Zhou, Y., Li, S., & Bogdan, P. (2024). Exploring neuron interactions and emergence in large language models: A multifractal analysis perspective. *arXiv preprint*, arXiv:2402.09099.
- Cheng, M., Li, Y., Nazarian, S., & Bogdan, P. (2021). From rumor to genetic mutation detection with explanations: A GAN approach. *Scientific Reports*, 11, 5861.
- Cheng, M., Yin, C., Nazarian, S., & Bogdan, P. (2021). Deciphering the laws of social network-transcendent COVID-19 misinformation dynamics and implications for combating misinformation phenomena. *Scientific Reports*, 11, 10424.
- Li, H., Wang, W., Liu, Z., Niu, Y., Wang, H., Zhao, S., Liao, Y., Yang, W., & Liu, X. (2022). A novel locality-sensitive hashing relational graph matching network for **Semantic Textual Similarity** measurement. *Expert Systems with Applications*, 207, 117832.
- Almuhaimeed, A., Alhomidi, M. A., Alenezi, M. N., Alamoud, E., & Alqahtani, S. (2022). A modern semantic similarity method using multiple resources for enhancing influenza detection. *Expert Systems with Applications*, 193, 116466.
- Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & Nobani, N. (2022). Embeddings evaluation using a novel measure of **Semantic Similarity**. *Cognitive Computation*, 14, 749–763.
- Guo, W., Zeng, Q., Duan, H., Ni, W., & Liu, C. (2021). Process-extraction-based text similarity measure for emergency response plans. *Expert Systems with Applications*, 183, 115301.
- Lu, W., Zhang, X., Lu, H., & Li, F. (2020). Deep hierarchical encoding model for sentence semantic matching. *Journal of Visual Communication and Image Representation*, 71, 102794.
- Kleenankandy, J., & A. N. K. A. (2020). An enhanced Tree-LSTM architecture for sentence semantic modeling using typed dependencies. *Information Processing & Management*, 57, 102362.
- Oussalah, M., & Mohamed, M. (2021). Knowledge-based sentence semantic similarity: Algebraical properties. *Progress in Artificial Intelligence*, 11, 43–63.
- Meshram, S., & Anand Kumar, M. (2021). Long short-term memory network for learning sentence similarity using deep contextual embeddings. *International Journal of Information Technology*, 13, 1633–1641.
- Miller, G. A. (1994). WordNet: A lexical database for English. In *Proceedings of the Human Language Technology Workshop*, Plainsboro, NJ, USA.
- Bhattacharyya, P. (2010). IndoWordNet. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.
- Panjwani, R., Kanojia, D., & Bhattacharyya, P. (2018). pyiwn: A Python-based API to access Indian language WordNets. In *Proceedings of the Global WordNet Conference* (pp. 378–383).

-
- Pawar, A., & Mago, V. K. (2018). Calculating similarity between words and sentences using a lexical database and corpus statistics. *arXiv preprint*, arXiv:1802.05667.
- Jurafsky, D., & Martin, J. H. (2013). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson, Upper Saddle River, NJ, USA.
- Iacobacci, I., Pilehvar, M. T., & Navigli, R. (2016). Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 897–907).
- Kumar, S., Kumar, S., Kanojia, D., & Bhattacharyya, P. (2020). "A passage to India": Pre-trained word embeddings for Indian languages. In *Proceedings of the Joint Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU) and CCURL* (pp. 352–357).
- Ruder, S., Vulić, I., & Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65, 569–630.
- Honnibal, M., & Montani, I. (2018). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *Sentometrics Research*, 7, 411–420.
- Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146.
- Artetxe, M., Labaka, G., & Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 2289–2294).
- Artetxe, M., Labaka, G., & Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 451–462).
- Artetxe, M., Labaka, G., & Agirre, E. (2018). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 5012–5019).
- Artetxe, M., Labaka, G., & Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 789–798).
- Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Hernandez Abrego, G., Yuan, S., Tar, C., & Sung, Y. H. (2020). Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 87–94).
- Smith, S. L., Turban, D. H. P., Hamblin, S., & Hammerla, N. Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Liu, X., Zhou, Y., & Zheng, R. (2007). Sentence similarity based on dynamic time warping. In *Proceedings of the International Conference on Semantic Computing (ICSC)* (pp. 250–256).
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Sebastopol, CA, USA.
- Pvs, A., & Gali, K. (2007). Part-of-speech tagging and chunking using conditional random fields and transformation-based learning. In *Proceedings of the Shallow Parsing for South Asian Languages (SPSAL) Workshop* (pp. 21–24).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 6000–6010).
- Wang, B., Zhao, D., Lioma, C., Li, Q., Zhang, P., & Simonsen, J. G. (2020). Encoding word order in complex embeddings. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
-

-
- Ramesh, G., Doddapaneni, S., Bheemaraj, A., Jobanputra, M., AK, R., Sharma, A., Sahoo, S., Diddee, H., J, M., Kakwani, D., et al. (2022). Samanantar: The largest publicly available parallel corpora collection for 11 Indic languages. *Transactions of the Association for Computational Linguistics*, 10, 145–162.
- Muralikrishna S N ,Raghurama Holla ,Harivinod N , andRaghavendra Ganiga 4:Cross-Lingual Short-Text Semantic Similarity for Kannada–English Language Pair Journals Computers Volume 13 Issue 9 10.3390/computers13090236