computers

MDPI

Article

Cross-Lingual Short-Text Semantic Similarity for Kannada–English Language Pair

Muralikrishna S N [1,†], Raghurama Holla [2,*,†], Harivinod N [3,†] and Raghavendra Ganiga [4,†]

[1] Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India; murali.sn@manipal.edu
[2] Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India
[3] Department of Computer Science and Engineering, St Joseph Engineering College, Mangaluru 575028, India; harivinodn@sjec.ac.in
[4] Department of Information & Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India; raghavendra.n@manipal.edu
* Correspondence: raghuram.holla@manipal.edu
† These authors contributed equally to this work.

Abstract: Analyzing the semantic similarity of cross-lingual texts is a crucial part of natural language processing (NLP). The computation of semantic similarity is essential for a variety of tasks such as evaluating machine translation systems, quality checking human translation, information retrieval, plagiarism checks, etc. In this paper, we propose a method for measuring the semantic similarity of Kannada–English sentence pairs that uses embedding space alignment, lexical decomposition, word order, and a convolutional neural network. The proposed method achieves a maximum correlation of 83% with human annotations. Experiments on semantic matching and retrieval tasks resulted in promising results in terms of precision and recall.

Keywords: sentence similarity; word embedding; Kannada monolingual embedding; short-text semantic similarity; lexical decomposition; cross-lingual semantic similarity

check for updates

Citation: S N, M.; Holla, R.; N, H.; Ganiga, R. Cross-Lingual Short-Text Semantic Similarity for Kannada–English Language Pair. *Computers* **2024**, *13*, 236. https://doi.org/10.3390/computers13090236

Academic Editor: Paolo Bellavista

Received: 11 July 2024
Revised: 14 August 2024
Accepted: 26 August 2024
Published: 18 September 2024

1. Introduction

The fundamental and difficult task in the field of natural language processing is determining the semantic similarity between two sentences. In the past several years, a significant amount of textual data have been produced and made available via websites, online news articles, etc. So, there is a need to design an effective method to calculate the semantic similarity between text data present in the given input documents. It has applications in many NLP domains, including text summarization, information retrieval, text categorization, machine translation, question-answering systems, and webpage retrieval.

For the calculation of semantic similarity, early methods used techniques like Term Frequency–Inverse Document Frequency (TF-IDF), Bag of Words (BoW), etc., to describe text data. However, certain words have many meanings depending on the context, and the same notion may be expressed by a different collection of words; therefore, these strategies may fail. Think about the phrases "Tony and Peter studied math and physics" and "Tony studied math and Peter studied physics" for instance. Despite the fact that these two sentences contain the exact same words, their meanings are completely different. The two sentences "John loves athletics" and "John is fond of sports" have a similar structure but use different terms to communicate the same idea. The older approaches thus ignore the semantic characteristics of the text and only capture its lexical features. Semantic-based similarity approaches have been suggested as a way to get around these restrictions. Semantic textual similarity (STS) measures the semantic relationship between text data by computing a rank or percentage of similarity between texts.

Estimating the semantic similarity between two sentences in various languages is the basic goal of cross-lingual semantic textual-similarity systems. This is a difficult challenge in natural language processing as well because it compares things like topics, emotions, and opinions [1–4]. In this work, we propose a method to calculate cross-lingual semantic similarity for the Kannada language, which is spoken by more than 60 million people in the southern Indian state of Karnataka. Moreover, the All India Council for Technical Education (AICTE) has started the process of translating educational course materials and textbooks into many Indian languages, including Kannada. As a result, India has a huge market for cross-lingual semantic-similarity systems. Figure 1 illustrates an instance of a cross-lingual semantic-similarity calculation in a practical application. In this scenario, a module designed for cross-lingual semantic similarity assesses the similarity between an English answer key and a Kannada response. This approach can streamline answer paper evaluation, especially in multilingual countries where responses are often written in local languages. Assuming digitized and accessible responses, the module's score can feed into a decision system. India, with its over 22 official languages, has increasingly promoted local language use in education and exams. While questions are often provided in English and other local languages, responses can be in any language. By comparing the English answer key to the local language response, semantic similarity can be assessed. This method holds potential for machine translation evaluation and automatic grading. The following are the main contributions of the proposed research work:

- We provide a method for computing the semantic textual similarity (STS) between sentences in Kannada and English that utilizes lexical decomposition, embedding space alignment, and convolutional neural networks. To the best of our knowledge, this is the first attempt to measure STS for the Kannada–English language pair.
- We assess the proposed method's performance in terms of precision and correlation with the human-annotated scores, as well as word-level alignment in the embedding space.

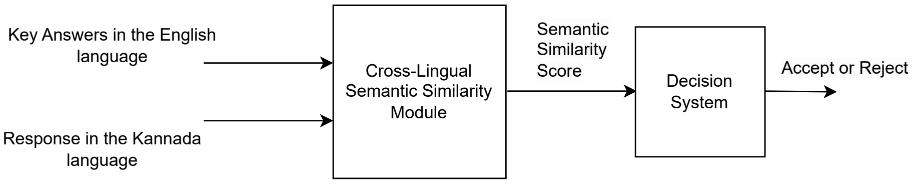

Key Answers in the English language
Response in the Kannada language
Cross-Lingual Semantic Similarity Module
Semantic Similarity Score
Decision System
Accept or Reject

Figure 1. Cross-lingual semantic-similarity calculation in multilingual-answer paper evaluation.

The remaining part of this paper is structured as follows: In Section 2, the related works on semantic-similarity estimation are discussed. The proposed work is presented in Section 3. In Section 4, the experimental results are evaluated and analyzed. Finally, we conclude this paper in Section 5.

2. A Review of Existing Works

It was discovered that there are various methods for determining the semantic similarity of sentences in the literature [5–10]. Three key elements that influence the calculation of semantic similarity were found in [11]. They are word embedding, word weighting measures, and similarity measures. These factors are analyzed using clustering algorithms. In a different strategy [12], the authors looked at lexical, syntactic, and semantic features to identify semantic similarity between phrases in Arabic. To find semantic similarity, machine learning algorithms like linear regression and SVM are used.

To handle uncertainty in the text data, the authors of [13] presented an enhanced approach based on the probabilistic tolerance rough set model. To assess the overall sentence similarity, the approach combines two measures: lower approximation similarity and upper approximation similarity. The authors of [14] proposed a method for determining sentence similarity that is based on a tree kernel and is known as the ACVT kernel. Syntactic data, semantic traits, and an attention-weighting mechanism are all combined. A strategy based on Fuzzy Rough Set was used to calculate sentence similarity in sentences because

Fuzzy Rough Set can handle uncertainty, imprecision, and incompleteness [15]. In another study [16,17], the authors put forth a method for determining text similarity based on word embedding, with an estimation of sentence structure similarity made using graph representations. In order to increase the precision of the sentence-similarity calculation, a mixture of many similarity measurements is employed. Alternatively, in [18], the researchers proposed a technique that leverages word embeddings and external knowledge sources to estimate semantic similarity in short text. In this approach, each short text is represented as two dense vectors. In another work [19], a model for semantic similarity has been proposed based on Optimal Alignment Similarity (OptAlign), Greedy Association Similarity (GrAssoc), and Aggregation Similarity (Aggreg). OptAlign uses optimal word alignment to identify equivalent words across texts. GrAssoc utilizes a greedy algorithm in conjunction with bilingual word embeddings to determine the corresponding words between two text sequences. Aggreg calculates a unified STS score by merging various similarity metrics using an aggregation function. The authors of [20] proposed an embedding-based topic model to effectively identify latent topics from short texts. This approach aggregates the short texts into longer 'pseudo-texts' and then uses a Markov Random Field to infer the topics from the pseudo-texts.

Deep-learning-based models were demonstrated to perform better than other models for NLP tasks [21–24]. A CNN-based approach was employed in [25] to determine the semantic similarity of sentences in medical documents. Recurrent Neural Networks (RNNs) are being commonly employed in Siamese neural networks [26,27] to calculate the degree of similarity between pairs of texts. The recurrent convolutional neural network (RCNN) [28] is another technique used to learn semantic similarity between two sentences. Siamese multilayer CNNs were employed by the authors to extract key information from the input phrases. After combining several representations of the input sentences using a fusion layer, the final similarity score is determined.

To increase the accuracy of their sentence-similarity computation, the authors took a different approach [29] and created a three-layered model made up of (i) a lexical layer, (ii) a syntactic layer, and (iii) a semantic layer. To evaluate the semantic similarity in texts, the authors in [30] developed an attention model made up of word N-grams produced by a bi-directional RNN. In [31], the authors introduced an algorithm that takes into account the similar component and the dissimilar component of every word based on a semantic matching vector and then computes a similarity score in order to address various issues in semantic-similarity computation.

By performing sentence-by-sentence alignment, a different technique was suggested in [32] to determine the semantic textual similarity between sentences. The multiple linear regression method was used to determine an alignment score. SVM, Gaussian Naive Bayes, and KNN classifiers were used via the classification method to conduct multiclass classification. In [33], to conduct a semantic-similarity study between two lines or paragraphs in Bengali, a combination of six statistical parameters, including frequency, matching index, part of speech, entropy, cosine similarity, and sense matching was used. A hybrid strategy was presented in another work [34] by extracting features using the Weighted Fine-tuned BERT algorithm. The Siamese Bi-LSTM network model was used to train the feature vectors. Finally, the similarity scores for each sentence were determined.

Using a deep model built on a bi-directional interaction network, the authors of [35] were able to accurately capture the semantic relationship between sentences. The model combines a deep fusion layer, a deep neural network, and an attention mechanism. Multilayer perceptrons were used for the classification. We may also observe a gated network-based approach [36] for estimating semantic similarity that combines distributed representation and one-hot representation. In [37], the authors introduced the *W-KG2Vec* meta-path-specific model to capture the semantic relationship as well as the text similarity between phrases. Their suggested model BERT-Text2Vec combines an LSTM encoder and a BERT pre-trained model. Along with word-embedding representations, sentiment lexicons and semantic models are also used [38] to detect semantic similarity in texts. In [39], the authors suggested an approach to measure text similarity based on the combination of syntactic and semantic information using knowledge and Corpora. In another work [40], the authors presented KNetwork, an innovative approach for cross-lingual sentiment analysis; this method employs feature vectors derived from translated and transliterated text to enhance accuracy in various linguistic contexts. In [41], the authors introduced a novel approach that combines various features, incorporating two coarse-grained prior-knowledge aspects (sentence sentiment and length), and leverages the XLM-RoBERTa cross-lingual pre-training model. The paper underscores the importance of incorporating both fine-grained and coarse-grained features in sentence-similarity computations to address language-specific biases and enhance overall accuracy. In [42], the authors proposed an approach for the sentiment analysis of Sanskrit texts, addressing the difficulty arising from the limited availability of labeled data for training machine learning models. The methodology relies on zero-shot cross-lingual mapping, enabling sentiment classification in Sanskrit without direct sentiment labels by transferring knowledge from English sentiment labels. In another approach [43], the authors proposed a framework that explains LLMs by analyzing the patterns of communication between neurons within the model. In [44], the researchers developed a GAN-based model, which includes a generative model and a discriminative model, for the purpose of detecting rumors or fake news. In another work [45], the researchers developed a misinformation network model to detect misinformation and fake information during the COVID-19 pandemic.

We observe that several approaches are available for sentence-similarity calculation, and some of the recent approaches can be seen in [46–53]. However, very limited approaches are available in the literature for calculating the semantic similarity between the Kannada and English languages. In this direction, we made an attempt to develop a cross-lingual semantic-similarity approach for Kannada–English sentence pairs. The details of the proposed approach are discussed in Section 3.

3. Methodology

The current short-text semantic similarity (STS) methods find similarity scores between two sentence pairs using word-based or vector-based approaches. The word-based similarity measure uses a lexical database to compute the semantic similarity between a pair of words from two sentences. The semantic equivalency of the word pairings from the two sentences, as well as the alignment of the texts in the two sentences, are used to determine the similarity score. Word embedding, a deep neural network (DNN)-based technique, is used to encode a word into a vector form in a vector-based representation. A big corpus of data was used to train the DNN to produce this vector representation. Thus, the representation can capture word dependencies, context, and semantics. Therefore, combining the two aforementioned strategies would enhance performance. In the proposed research work, we employ a hybrid methodology to investigate the benefits of a word-based methodology and vector-based methodology for computing the cross-lingual short-text semantic similarity on Kannada–English sentence pairs. We go into further detail on word-based similarity and vector-based similarity in the section that follows.

3.1. Word-Based Similarity vs. Vector-Based Similarity

A lexical database similar to WordNet [54–56] can be used to measure the similarity. WordNet is a hierarchical database where the cognitive synonyms (synset) are provided for nouns, verbs, adjectives, and adverbs as separate groups. In most of the existing word-based methods, as a first step, the sentences S and T are subjected to part-of-speech (POS) tagging. In the next step, each word is associated with a sense using *word sense disambiguation* (WSD). This helps to capture the contextual information of words. For example, the word *bank* may have different senses in different contexts. Considering the 'context' of a word helps with the optimal overall similarity calculation. It helps to use WordNet for noun–noun and verb–verb pairs for two words $s_i \in S$ and $t_j \in T$ to obtain the best alignment of words in two sentences. The path length in WordNet between the words and their depth in the tree is considered as the score for alignment [57]. The major advantage of using the word-similarity score with the lexical database is that it provides a mechanism to handle out-of-vocabulary words.

In vector-based representation, normally, a sentence S is represented using a vector $[s_1, s_2, ...s_m]$. In vector-based representation, each word s_i is represented with a pre-trained word embedding. Most of the vector-based methods obtain a similarity score between two short sentences S and T by measuring the similarity between the word pairs (s_i, t_j) in both sentences with the best matching word pairs. However, these methods fail to keep track of the word order and also fail to handle out-of-vocabulary (OOV) words. It is very common to observe OOV words in domain-specific applications. For example, in natural language processing (NLP) applications for healthcare, you are more likely to see common words in a different order that indicate symptoms, signs, diseases, etc. The word order is important to capture the syntactic structure of the sentence. For example, *'Your dog chased my cat'* vs. *'My cat chased your dog'*. The parse trees for these two sentences are not identical. So, it is necessary to consider the word order in sentence-similarity-score computation. The English language follows a strict word order but allows for some freedom at the phrase level [58]. The proposed architecture for semantic-similarity computation is shown in Figure 2. The detailed architecture is shown in Figure 3. The sentence S and sentence T are from two different language pairs. In our case, we perform semantic similarity between Kannada and English short-sentence pairs. As a pre-processing step, we perform tokenization, stopword removal, and word sense disambiguation (WSD). Though the word embedding can capture the context, we apply WSD to English words to enhance the overall performance [59]. The word embedding is carried out as explained in Section 3.2.

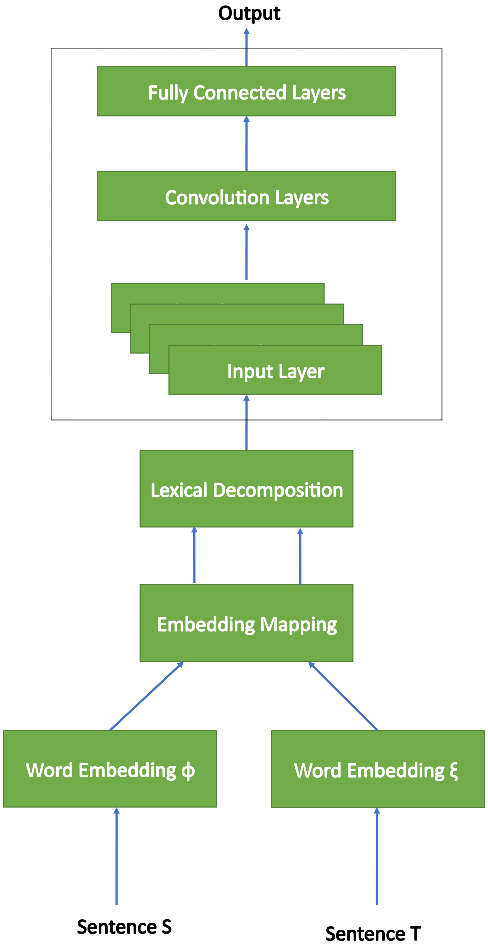


Figure 2. Proposed architecture for semantic-similarity computation.

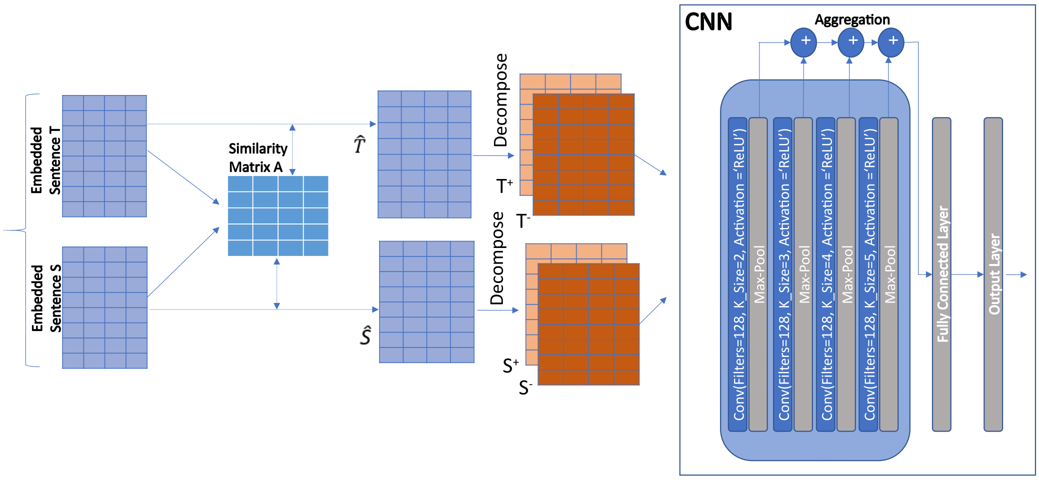

Embedded Sentence T
Embedded Sentence S
Similarity Matrix A
$\hat{T}$
$\hat{S}$
Decompose
Decompose
T+
T-
S+
S-
CNN
Aggregation
+
+
+
Conv(Filters=128, K_Size=2, Activation ='ReLU')
Max-Pool
Conv(Filters=128, K_Size=3, Activation ='ReLU')
Max-Pool
Conv(Filters=128, K_Size=4, Activation ='ReLU')
Max-Pool
Conv(Filters=128, K_Size=5, Activation ='ReLU')
Max-Pool
Fully Connected Layer
Output Layer

Figure 3. A detailed architecture for cross-lingual semantic-similarity computation.

3.2. Word Embedding

We use a word-embedding scheme to create a vector representation of each word appearing in a sentence. In word embedding, a d-dimensional vector is used to represent each word $s_i \in S$ or $t_j \in T$ with a pre-trained neural network [60]. Some of the commonly used embedding schemes are *Word2Vec*, *GloVe*, *FasText*, etc. The *Word2Vec* uses skip-gram or the *Continuous Bag of Word model* (CBOW) for the embedding. A d-dimensional embedding process generates a matrix of dimension $n \times d$ for a sentence with n words. We apply this scheme to both sentences $S = [s_1, s_2, \ldots s_m]$ and $T = [t_1, t_2, \ldots, t_n]$ to obtain the word embedding of both sentences in matrix form. Each s_i and t_j represents the word-embedding vector in d-dimensional space. The pre-trained embedding models can be of two types: (i) the monolingual embedding and (ii) the cross-lingual embedding. In monolingual embedding, the model is trained on a dataset containing text from a single language, whereas in cross-lingual embedding, a dataset containing a parallel corpus with or without word alignment is used to train the model [61]. In our experiment, we explore both monolingual embedding and cross-lingual embedding for sentence representations. The processed texts are embedded into a 300 d vector using pre-trained word-embedding models. The English sentences are embedded using the *tok2vec* model available in Spacy library [62]. Similarly, the Kannada text is embedded using *fasttext* [63]. It uses the skip-gram model with default parameters as described in [64] for Kannada word embedding (the pre-trained word embedding for Kannada is available at https://fasttext.cc/docs/en/pretrained-vectors.html (accessed on 10 June 2024). The method presented in [64] uses a subword representation using character n-grams, thereby enabling the efficient representation of morphologically rich texts.

Monolingual Embedding and Alignment

As we use two different monolingual embeddings for English and Kannada, we perform the alignment of the two embeddings using *VecMap* and MUSE [65–69]. The *VecMap* and MUSE can align the embedding space using supervised or unsupervised methods. The supervised method uses a bilingual dictionary for the alignment. The major concern is the availability of a bilingual dictionary for Kannada–English pairs. The available dictionary has approximately 4 K words, which is inadequate considering the morphologically rich, agglutinative nature of the Kannada language. So, alternatively, we use an unsupervised alignment using the method specified in [70] for creating aligned word embeddings (offline bilingual word vectors, source code available at https://github.com/babylonhealth/fastText_multilingual (accessed on 10 June 2024).

3.3. Lexical Decomposition and Similarity Calculation

From the word embedding stated in the earlier section, we construct a word-similarity matrix denoted by $A_{m\times n}$, where each element a_{ij} is the cosine similarity between the vectors s_i and t_j. The cosine similarity between two vectors is given by

$$cos(\theta) = \frac{s_i \cdot t_j}{|s_i||t_j|}. \tag{1}$$

The range of the cosine similarity varies from -1 to 1. A similarity value of zero indicates that the vectors are orthogonal to each other (highly dissimilar). A value of +1 indicates the highest similarity, and a value of -1 indicates that the vectors are opposite to each other.

This matrix is essential to find the best match for the word (or phrase) s_i in a sentence S with a word (or phrase) t_j in T. The process of finding a match has to be performed for words in both the sentences S and T. This can be formulated as

$$\hat{s_i} = \text{find-match}(s_i, T) = \frac{\sum_{j=0}^{n} a_{ij} * t_j}{\sum_{j=0}^{n} a_{i,j}}. \tag{2}$$

This formulation is similar to the method proposed in [31].

The accuracy of the semantic matching greatly depends on the dimension of the word embedding used and also the word-embedding scheme. The match gives a semantic coverage for the words s_i in T, and similarly for the words t_j in S. We generate the complementary information of semantic coverage by decomposing the vectors s_i and t_j. Here, we consider that the complementary information is represented by an orthogonal vector. So, the decomposition is performed by computing a parallel vector and orthogonal vector termed as *similar components* and *dissimilar components* s_i^+ and s_i^-:

$$\begin{aligned} s_i^+ &= \frac{s_i \cdot \hat{s}_i}{\hat{s}_i \cdot \hat{s}_i}\hat{s}_i \\ s_i^- &= s_i - s_i^+. \end{aligned} \tag{3}$$

Finally, a convolutional neural network (CNN) is used to generate a feature vector for each sentence based on similar and dissimilar components using the convolution operation. The architecture of the CNN is shown in Figure 3. The feature vector is generated using the multiple convolution operation with varying kernel sizes $d \times k$, where d is the embedding dimension and $k = 2, 3, 4, 5$. This operation is analogous to feature extraction using the *n-gram* model. The convolution operations are followed by max-pooling. Finally, the features are aggregated by concatenation. The final layers of the CNN are the fully connected layer and the output layer, which works as the classification head.

3.4. Word-Order Similarity

In the proposed method, we consider word-order similarity to be one of the factors to estimate the semantic similarity of sentences. Given two sentences with a sequence of words, $S = [s_1, s_2, \ldots, s_m]$ and $T = [t_1, t_2, \ldots, t_n]$, we use two strategies to include the word order in sentences. The first method is to compute the word-order similarity using Dynamic Time Warping (DTW) [71]. As DTW describes a distance measure and not a similarity measure of two sentences, we use $d(w_1, w_2) = 1 - sim(w_1, w_2)$ as the distance between two words w_1 and w_2, where $sim(w_1, w_2)$ is the cosine similarity measure of their word embedding. DTW is defined for real numbers, so we use the distance for DTW. The dynamic programming formulation of DTW is given in Equation (4):

$$DTW(S, T) = f(m, n) \tag{4}$$

$$f(m,n) = d(s_i,t_j) + min\begin{cases} f(i-1,j), \\ f(i,j-1), \\ f(i-1,j-1) \end{cases} \tag{5}$$

$$f(0,0) = 0 \quad f(i,0) = f(0,j) = \infty$$

$$where \quad i \in (0,m), \quad j \in (0,n);$$

$$D_\alpha(S,T) = \frac{DTW(S,T)}{length(DTW)}. \tag{6}$$

In Equation (6), $D_\alpha(S,T)$ denotes the word-order similarity score between the sentences S and T. We apply part-of-speech tagging [72,73] to give lesser importance to verbs and adverbs than the nouns and adjectives. We consider the noun, pronoun, and adjective sequences of words denoted by S_p or T_p and the verb, adverb, and other sequences of words denoted by S_q or T_q separately for each sentence. Now, the similarity score considering word order is formed by the weighted sum as given in Equation (7). This helps to handle the structural difference in Kannada–English sentence formation. The proposed vector representation does not consider the function words. Thus, they are removed in the pre-processing stage:

$$D_{wo} = \gamma * D_\alpha(S_p,T_p) + (1-\gamma) * D_\alpha(S_q,T_q). \tag{7}$$

Alternatively, we use positional encoding during the embedding to incorporate the word order. This is similar to the technique of the self-attention mechanism used in transformers [74,75]. The positional encoding (PE) can be summarized using Equation (8):

$$\begin{aligned} PE_{(pos,2*i)} &= \sin(pos/10{,}000^{2i/d_{model}}) \\ PE_{(pos,2*i+1)} &= \cos(pos/10{,}000^{2i/d_{model}}) \end{aligned} \tag{8}$$

where pos represents the position of the word in the sequence and i indicates the dimension. The term d_{model} is the dimension of the word embedding used. The positional encoding is added to the word embedding to obtain the final input vectors.

3.5. Score-Level Fusion

The CNN architecture with minor modifications is used to incorporate DTW-based word-order similarity. The score-level fusion is achieved using an additional node with logistic regression. The inputs to this node are the vector-based similarity score and the word-order similarity. The cost function for this logistic regression is defined in Equation (9). The degree of the regression model used is two:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{2}[-y_i log(h_\theta(z_i)) - (1-y_i)log(1-h_\theta(z_i))] \tag{9}$$

where the term y_i indicates the expected output for the ith sample and $h_\theta(z_i)$ is the predicted value for the same sample.

4. Experimentation and Results

We use the TensorFlow library for the implementation of the proposed approach. The implemented code is run on an 11th Gen Intel(R), Core-i5, 2.70 GHz Processor with 16 GB RAM and NVIDIA GeForce RTX 3050 GPU having 8 GB memory. Our pre-processing tasks involve steps, like eliminating stop words, tagging parts of speech-disambiguating word senses, and filtering words using WordNet. Our implementation source code is available at https://github.com/muralikrishnasn/Kannada_English_STS (accessed on 10 July 2024). In our experiments, we use the Samanantar Parallel Corpora Dataset [76] to evaluate the performance of the proposed method. The Samanantar Corpora (https://ai4bharat.iitm

.ac.in/samanantar (accessed on 11 June 2024) is available for pairs of sentences in English and 11 Indic languages. The sources of this dataset are the WAT 2021 MultiIndicMT shared task, OPUS, and other sources. The dataset comprises 49.7 M parallel sentences between English and 11 Indic languages, out of which 12.4 M are taken from the existing sources. The remaining 37.4 M sentences are scraped from various online content. This shows that the data used for experimentation are from diverse domains. During dataset preparation, to determine the sentence pair from the scrapped text, the LabSE score is used. Also, a subset of the collected data were manually annotated for semantic sentence similarity (STS) into three classes, namely *definitely accept*, *marginal accept*, and *reject*, as shown in Figure 4. The annotation was performed by sampling 9.566 K English–Indic sentence pairs across 11 Indic languages. The Kannada–English pair has 0.957 K annotated sentences, out of which 70% are used for training and the remaining samples are used for testing.

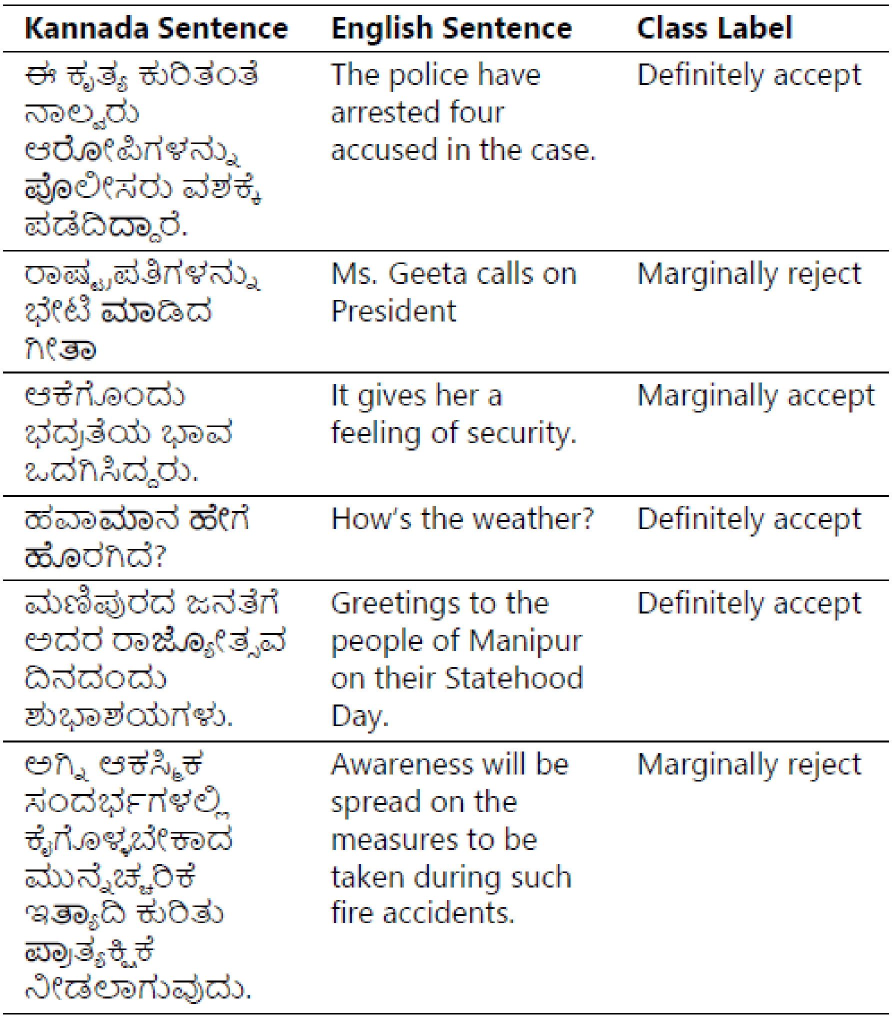
To evaluate the performance of STS and assess embedding alignment, we used a dataset containing human-annotated ground truth and a bilingual dictionary for alignment. We treated embedding alignment as a retrieval task and tabulated the results of retrieving semantically equivalent words from the target language for each source-language word in Table 1. The cosine similarity between their embedding representations is used to measure the precision at different ranks denoted by P@1, P@5, and P@10. Here, P@K represents the K number of words considered to calculate the precision in retrieval. We follow the assessment similar to [19]. We empirically fixed the CNN parameters as follows:

activation function = ReLU,
epochs = 100 ,
learning rate = 0.001,
$\epsilon = 1 \times 10^{-7}$,
loss function = cross-entropy.

Table 1. Assessment of embedding alignment during retrieval at different ranks denoted by @K.

Type of Learning	Alignment Method	Source–Target Pair	P@1	P@5	P@10
Unsupervised	VecMap	KAN - ENG	42.3	72.5	76.2
	MUSE	KAN - ENG	36.9	70.1	78.3
Supervised	VecMap	KAN - ENG	49.6	73.7	80.3
	MUSE	KAN - ENG	47.1	74.3	83.7

No substantial progress has been made in the domain of Kannada–English short-text semantic similarity, making the proposed work one of a kind. We conducted a comparative study using state-of-the-art techniques [19]: *OptAlign*, *GrAssoc*, and *Aggreg*, and the results are presented in Table 2. However, it is evident from the literature that only a few methods have been developed for research on cross-lingual semantic similarity between Kannada and English sentences. The performance of STS at the sentence level is measured in terms of the Spearman correlation coefficient (in % @rank-5) with the human-annotated labels for 4 K sentences. Also, we measure the Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP) to evaluate the retrieval task. Finally, we assess the effectiveness of the proposed method for semantic similarity considering word-embedding alignment with different bilingual dictionary sizes. In Figure 5, we can observe that the word-embedding alignment has no significant improvement beyond the 4 K bilingual dictionary. Also, we conducted a comparative analysis of cross-lingual STS performance on word order using positional embedding and DTW, as shown in Figure 6.



Kannada Sentence	English Sentence	Class Label
ಈ ಕೃತ್ಯ ಕುರಿತಂತೆ ನಾಲ್ವರು ಆರೋಪಿಗಳನ್ನು ಪೊಲೀಸರು ವಶಕ್ಕೆ ಪಡೆದಿದ್ದಾರೆ.	The police have arrested four accused in the case.	Definitely accept
ರಾಷ್ಟ್ರಪತಿಗಳನ್ನು ಭೇಟಿ ಮಾಡಿದ ಗೀತಾ	Ms. Geeta calls on President	Marginally reject
ಆಕೆಗೊಂದು ಭದ್ರತೆಯ ಭಾವ ಒದಗಿಸಿದ್ದರು.	It gives her a feeling of security.	Marginally accept
ಹವಾಮಾನ ಹೇಗೆ ಹೊರಗಿದೆ?	How's the weather?	Definitely accept
ಮಣಿಪುರದ ಜನತೆಗೆ ಅದರ ರಾಜ್ಯೋತ್ಸವ ದಿನದಂದು ಶುಭಾಶಯಗಳು.	Greetings to the people of Manipur on their Statehood Day.	Definitely accept
ಅಗ್ನಿ ಆಕಸ್ಮಿಕ ಸಂದರ್ಭಗಳಲ್ಲಿ ಕೈಗೊಳ್ಳಬೇಕಾದ ಮುನ್ನೆಚ್ಚರಿಕೆ ಇತ್ಯಾದಿ ಕುರಿತು ಪ್ರಾತ್ಯಕ್ಷಿಕೆ ನೀಡಲಾಗುವುದು.	Awareness will be spread on the measures to be taken during such fire accidents.	Marginally reject

Figure 4. A few sample texts from the dataset.

Table 2. Comparative analysis of experimental results with other methods given in [19] on Samanantar Parallel Corpora Dataset [76] using supervised MUSE alignment.

Model	MAP	MRR	Spearman Correlation Coefficient
OptAlign	0.71	0.82	0.75
GrAssoc	0.68	0.64	0.71
Aggreg	0.59	0.62	0.62
Proposed Method	0.81	0.85	0.83

VecMap and MUSE are both powerful tools for cross-lingual word embeddings, but they have different approaches and features. Based on the findings in Tables 1 and 2, it is evident that determining the superiority between VecMap and MUSE is not straightforward. The choice between these methods relies on the specific demands and limitations of our embedding-space-alignment task. Nonetheless, it is worth noting that MUSE demonstrates an improved performance for the KAN-ENG language pair when equipped with a dictionary containing more than 2 K word pairs, as shown in Figure 5. Also, VecMap requires bilingual dictionaries or parallel data and has broader language support, while MUSE can be trained with monolingual data alone and has demonstrated a competitive performance in various tasks.

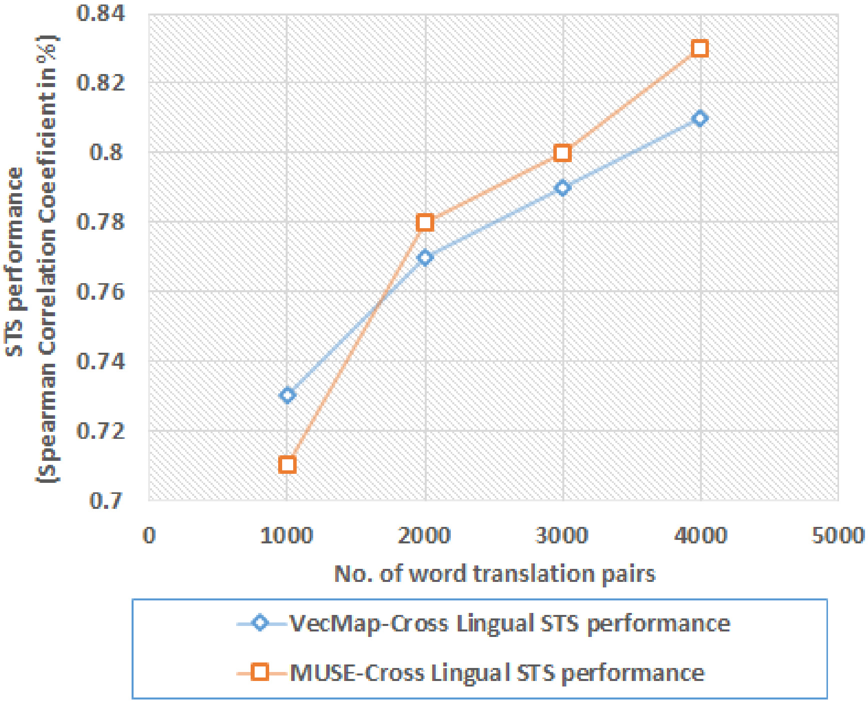


Figure 5. Cross-lingual STS performance.

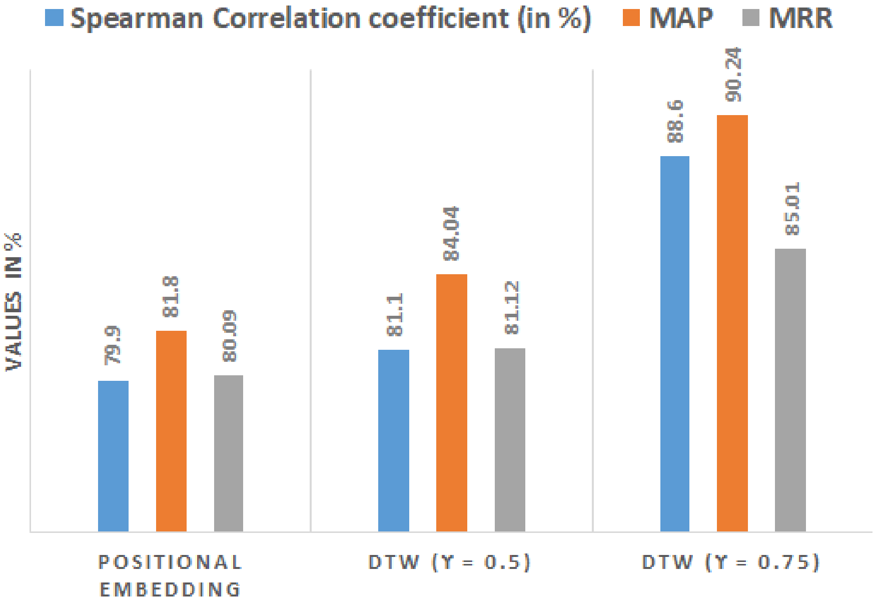


Figure 6. A comparison of cross-lingual STS performance considering word order using positional embedding and DTW.

5. Conclusions

We developed an approach to calculate semantic similarity between two sentences based on embedding space alignment, lexical decomposition, and a CNN. The process of embedding alignment with supervised MUSE performs well for Kannada–English language pairs, even though the Kannada language is morphologically rich and agglutinative. Lexical decomposition and word-order analysis using the DTW mechanism enable the capture of additional semantics in the sentences, which leads to an improved overall performance as measured by the correlation with human annotations. The proposed method also highlights the significance of the bilingual dictionary in achieving a better performance. Developing a lexical database for the Kannada language would facilitate the development of methods to handle semantic-matching-related tasks such as machine translation, information retrieval, and sentiment analysis.

In our future research, we plan to explore the effectiveness of different embedding schemes and different alignment schemes on Kannada–English language pairs for STS. We will also extend the work for other Indian language pairs by using architectures based on modified CNNs. This will allow the model to focus on more relevant parts of the sentences for improved performance. Finally, we are curious to examine the impact of the proposed method on diverse datasets for cross-lingual STS tasks.

Author Contributions: Conceptualization, M.S.N.; methodology, M.S.N.; software, M.S.N.; validation, H.N.; formal analysis, H.N.; investigation, H.N. and R.H.; resources, M.S.N.; writing—original draft preparation, R.H.; writing—review and editing, R.G.; visualization, R.G.; supervision, H.N.; project administration, H.N. and R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are openly available in the Samanantar Parallel Corpora Dataset [76], which can be accessed at https://ai4bharat.iitm.ac.in/samanantar (accessed on 11 June 2024).

Acknowledgments: We are thankful to the Manipal Academy of Higher Education for supporting this research work.

Conflicts of Interest: The authors declare no conflicts of interest regarding the publication of this article.

References

1. Pikuliak, M.; Šimko, M.; Bieliková, M. Cross-lingual learning for text processing: A survey. *Expert Syst. Appl.* **2021**, *165*, 113765. [CrossRef]
2. Saad, M.; Langlois, D.; Smaïli, K. Cross-Lingual Semantic Similarity Measure for Comparable Articles. In *Advances in Natural Language Processing*; Springer International Publishing: Cham, Switzerland, 2014; pp. 105–115.
3. Cer, D.M.; Diab, M.T.; Agirre, E.; Lopez-Gazpio, I.; Specia, L. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In Proceedings of the SemEval@ACL, Vancouver, BC, Canada, 3–4 August 2017.
4. Camacho-Collados, J.; Pilehvar, M.T.; Collier, N.; Navigli, R. SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity. In Proceedings of the SemEval@ACL, Vancouver, BC, Canada, 3–4 August 2017.
5. Zhao, C.; Wu, M.; Yang, X.; Zhang, W.; Zhang, S.; Wang, S.; Li, D. A Systematic Review of Cross-Lingual Sentiment Analysis: Tasks, Strategies, and Prospects. *ACM Comput. Surv.* **2024**, *56*, 1–37. [CrossRef]
6. Xu, Y.; Cao, H.; Du, W.; Wang, W. A Survey of Cross-lingual Sentiment Analysis: Methodologies, Models and Evaluations. *Data Sci. Eng.* **2022**, *7*, 279–299. [CrossRef]
7. Chandrasekaran, D.; Mago, V. Evolution of Semantic Similarity—A Survey. *ACM Comput. Surv.* **2021**, *54*, 41. [CrossRef]
8. Prakoso, D.W.; Abdi, A.; Amrit, C. Short text similarity measurement methods: A Review. *Soft Comput.* **2021**, *25*, 1–25. [CrossRef]
9. Navigli, R.; Martelli, F. An overview of word and sense similarity. *Nat. Lang. Eng.* **2019**, *25*, 693–714. [CrossRef]
10. Khattak, F.K.; Jeblee, S.; Pou-Prom, C.; Abdalla, M.; Meaney, C.; Rudzicz, F. A survey of word embeddings for clinical text. *J. Biomed. Inform.* **2019**, *100*, 100057. [CrossRef]
11. Alian, M.; Awajan, A. Factors affecting sentence similarity and paraphrasing identification. *Int. J. Speech Technol.* **2020**, *23*, 851–859. [CrossRef]
12. Alian, M.; Awajan, A. Arabic sentence similarity based on similarity features and machine learning. *Soft Comput.* **2021**, *25*, 10089–10101. [CrossRef]
13. Yan, R.; Qiu, D.; Jiang, H. Sentence Similarity Calculation Based on Probabilistic Tolerance Rough Sets. *Math. Probl. Eng.* **2021**, *2021*, 1–9. [CrossRef]
14. Quan, Z.; Wang, Z.J.; Le, Y.; Yao, B.; Li, K.; Yin, J. An efficient framework for sentence similarity modeling. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 853–865. [CrossRef]
15. Chatterjee, N.; Yadav, N. Fuzzy Rough Set-Based Sentence Similarity Measure and its Application to Text Summarization. *IETE Tech. Rev.* **2019**, *36*, 517–525. [CrossRef]
16. Kenter, T.; De Rijke, M. Short text similarity with word embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 1411–1420.
17. Farouk, M. Measuring text similarity based on structure and word embedding. *Cogn. Syst. Res.* **2020**, *63*, 1–10. [CrossRef]
18. Nguyen, H.T.; Duong, P.H.; Cambria, E. Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowl.-Based Syst.* **2019**, *182*, 104842. [CrossRef]

19. Glavaš, G.; Franco-Salvador, M.; Ponzetto, S.P.; Rosso, P. A resource-light method for cross-lingual semantic textual similarity. *Knowl.-Based Syst.* **2018**, *143*, 1–9. [CrossRef]
20. Qiang, J.; Chen, P.; Wang, T.; Wu, X. Topic Modeling over Short Texts by Incorporating Word Embeddings. In *Advances in Knowledge Discovery and Data Mining*; Kim, J., Shim, K., Cao, L., Lee, J.G., Lin, X., Moon, Y.S., Eds.; Springer: Cham, Switzerland, 2017; pp. 363–374.
21. Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 604–624. [CrossRef]
22. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2, Cambridge, MA, USA, 18–22 November 2014; pp. 2042–2050.
23. Hou, S.L.; Huang, X.K.; Fei, C.Q.; Zhang, S.H.; Li, Y.Y.; Sun, Q.L.; Wang, C.Q. A Survey of Text Summarization Approaches Based on Deep Learning. *J. Comput. Sci. Technol.* **2021**, *36*, 633–663. [CrossRef]
24. Alshemali, B.; Kalita, J. Improving the Reliability of Deep Neural Networks in NLP: A Review. *Knowl.-Based Syst.* **2020**, *191*, 105210. [CrossRef]
25. Zheng, T.; Gao, Y.; Wang, F.; Fan, C.; Fu, X.; Li, M.; Zhang, Y.; Zhang, S.; Ma, H. Detection of medical text semantic similarity based on convolutional neural network. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 156. [CrossRef]
26. Chicco, D. Siamese Neural Networks: An Overview. In *Artificial Neural Networks*; Humana: New York, NY, USA, 2021; pp. 73–94.
27. Ranasinghe, T.; Orasan, C.; Mitkov, R. Semantic Textual Similarity with Siamese Neural Networks. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), Varna, Bulgaria, 2–4 September 2019; pp. 1004–1011. [CrossRef]
28. Peng, S.; Cui, H.; Xie, N.; Li, S.; Zhang, J.; Li, X. Enhanced-RCNN: An Efficient Method for Learning Sentence Similarity. In Proceedings of the Web Conference 2020, New York, NY, USA, 20–24 April 2020; pp. 2500–2506.
29. Ferreira, R.; Lins, R.D.; Simske, S.J.; Freitas, F.; Riss, M. Assessing sentence similarity through lexical, syntactic and semantic analysis. *Comput. Speech Lang.* **2016**, *39*, 1–28. [CrossRef]
30. Lopez-Gazpio, I.; Maritxalar, M.; Lapata, M.; Agirre, E. Word n-gram attention models for sentence similarity and inference. *Expert Syst. Appl.* **2019**, *132*, 1–11. [CrossRef]
31. Wang, Z.; Mi, H.; Ittycheriah, A. Sentence Similarity Learning by Lexical Decomposition and Composition. In Proceedings of the COLING, Osaka, Japan, 11–16 December 2016.
32. Majumder, G.; Pakray, P.; Das, R.; Pinto, D. Interpretable semantic textual similarity of sentences using alignment of chunks with classification and regression. *Appl. Intell.* **2021**, *51*, 7322–7349. [CrossRef]
33. Das, A.; Mandal, J.; Danial, Z.; Pal, A.; Saha, D. A novel approach for automatic Bengali question answering system using semantic similarity analysis. *Int. J. Speech Technol.* **2020**, *23*, 873–884. [CrossRef]
34. Viji, D.; Revathy, S. A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi—LSTM model for semantic text similarity identification. *Multimed. Tools Appl.* **2022**, *81*, 1–27. [CrossRef]
35. Liu, M.; Zhang, Y.; Xu, J.; Chen, Y. Deep bi-directional interaction network for sentence matching. *Appl. Intell.* **2021**, *51*, 4305–4329. [CrossRef]
36. Xiong, Y.; Chen, S.; Qin, H.; Cao, H.; Shen, Y.; Wang, X.; Chen, Q.; Yan, J.; Tang, B. Distributed representation and one-hot representation fusion with gated network for clinical semantic textual similarity. *BMC Med. Inform. Decis. Mak.* **2020**, *20*, 72. [CrossRef]
37. Do, P.; Pham, P. W-KG2Vec: A weighted text-enhanced meta-path-based knowledge graph embedding for similarity search. *Neural Comput. Appl.* **2021**, *33*, 16533–16555. [CrossRef]
38. Araque, O.; Zhu, G.; Iglesias, C.A. A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowl.-Based Syst.* **2019**, *165*, 346–359. [CrossRef]
39. Yang, J.; Li, Y.; Gao, C.; Zhang, Y. Measuring the short text similarity based on semantic and syntactic information. *Future Gener. Comput. Syst.* **2021**, *114*, 169–180. [CrossRef]
40. Jain, A.; Jain, G.; Tewari, D. KNetwork: Advancing cross-lingual sentiment analysis for enhanced decision-making in linguistically diverse environments. *Knowl. Inf. Syst.* **2024**, *66*, 1–19. [CrossRef]
41. Wang, L.; Liu, S.; Qiao, L.; Sun, W.; Sun, Q.; Cheng, H. A Cross-Lingual Sentence Similarity Calculation Method with Multifeature Fusion. *IEEE Access* **2022**, *10*, 30666–30675. [CrossRef]
42. Kumar, P.; Pathania, K.; Raman, B. Zero-shot learning based cross-lingual sentiment analysis for sanskrit text with insufficient labeled data. *Appl. Intell.* **2023**, *53*, 10096–10113. [CrossRef]
43. Xiao, X.; Zhou, C.; Ping, H.; Cao, D.; Li, Y.; Zhou, Y.; Li, S.; Bogdan, P. Exploring Neuron Interactions and Emergence in LLMs: From the Multifractal Analysis Perspective. *arXiv* **2024**, arXiv:2402.09099.
44. Cheng, M.; Li, Y.; Nazarian, S.; Bogdan, P. From rumor to genetic mutation detection with explanations: A GAN approach. *Sci. Rep.* **2021**, *11*, 5861. [CrossRef] [PubMed]
45. Cheng, M.; Yin, C.; Nazarian, S.; Bogdan, P. Deciphering the laws of social network-transcendent COVID-19 misinformation dynamics and implications for combating misinformation phenomena. *Sci. Rep.* **2021**, *11*, 10424. [CrossRef]
46. Li, H.; Wang, W.; Liu, Z.; Niu, Y.; Wang, H.; Zhao, S.; Liao, Y.; Yang, W.; Liu, X. A Novel Locality-Sensitive Hashing Relational Graph Matching Network for Semantic Textual Similarity Measurement. *Expert Syst. Appl.* **2022**, *207*, 117832. [CrossRef]

47. Almuhaimeed, A.; Alhomidi, M.A.; Alenezi, M.N.; Alamoud, E.; Alqahtani, S. A modern semantic similarity method using multiple resources for enhancing influenza detection. *Expert Syst. Appl.* **2022**, *193*, 116466. [CrossRef]
48. Giabelli, A.; Malandri, L.; Mercorio, F.; Mezzanzanica, M.; Nobani, N. Embeddings Evaluation Using a Novel Measure of Semantic Similarity. *Cogn. Comput.* **2022**, *14*, 749–763. [CrossRef]
49. Guo, W.; Zeng, Q.; Duan, H.; Ni, W.; Liu, C. Process-extraction-based text similarity measure for emergency response plans. *Expert Syst. Appl.* **2021**, *183*, 115301. [CrossRef]
50. Lu, W.; Zhang, X.; Lu, H.; Li, F. Deep hierarchical encoding model for sentence semantic matching. *J. Vis. Commun. Image Represent.* **2020**, *71*, 102794. [CrossRef]
51. Kleenankandy, J.; K A, A.N. An enhanced Tree-LSTM architecture for sentence semantic modeling using typed dependencies. *Inf. Process. Manag.* **2020**, *57*, 102362. [CrossRef]
52. Oussalah, M.; Mohamed, M. Knowledge-based sentence semantic similarity: Algebraical properties. *Prog. Artif. Intell.* **2021**, *11*, 43–63. [CrossRef]
53. Meshram, S.; Anand Kumar, M. Long short-term memory network for learning sentences similarity using deep contextual embeddings. *Int. J. Inf. Technol.* **2021**, *13*, 1633–1641. [CrossRef]
54. Miller, G.A. WordNet: A Lexical Database for English. In Proceedings of the Human Language Technology: Proceedings of a Workshop, Plainsboro, NJ, USA, 8–11 March 1994.
55. Bhattacharyya, P. IndoWordNet. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 17–18 May 2010.
56. Panjwani, R.; Kanojia, D.; Bhattacharyya, P. pyiwn: A Python based API to access Indian Language WordNets. In Proceedings of the 9th Global Wordnet Conference, Nanyang Technological University (NTU), Singapore, 8–12 January 2018; pp. 378–383.
57. Pawar, A.; Mago, V.K. Calculating the similarity between words and sentences using a lexical database and corpus statistics. *arXiv* **2018**, arXiv:1802.05667.
58. Jurafsky, D.; Martin, J.H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2013.
59. Iacobacci, I.; Pilehvar, M.T.; Navigli, R. Embeddings for Word Sense Disambiguation: An Evaluation Study. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 897–907.
60. Kumar, S.; Kumar, S.; Kanojia, D.; Bhattacharyya, P. "A Passage to India": Pre-trained Word Embeddings for Indian Languages. In Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), Marseille, France, 11–12 May 2020; pp. 352–357.
61. Ruder, S.; Vulić, I.; Søgaard, A. A Survey of Cross-Lingual Word Embedding Models. *J. Artif. Int. Res.* **2019**, *65*, 569–630. [CrossRef]
62. Honnibal, M.; Montani, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *Sentometrics Res.* **2018**, *7*, 411–420.
63. Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; Mikolov, T. Learning Word Vectors for 157 Languages. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
64. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
65. Artetxe, M.; Labaka, G.; Agirre, E. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 2289–2294.
66. Artetxe, M.; Labaka, G.; Agirre, E. Learning bilingual word embeddings with (almost) no bilingual data. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 451–462.
67. Artetxe, M.; Labaka, G.; Agirre, E. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 5012–5019.
68. Artetxe, M.; Labaka, G.; Agirre, E. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 789–798.
69. Yang, Y.; Cer, D.; Ahmad, A.; Guo, M.; Law, J.; Constant, N.; Hernandez Abrego, G.; Yuan, S.; Tar, C.; Sung, Y.h.; et al. Multilingual Universal Sentence Encoder for Semantic Retrieval. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 5–10 July 2020; pp. 87–94.
70. Smith, S.L.; Turban, D.H.P.; Hamblin, S.; Hammerla, N.Y. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
71. Liu, X.; Zhou, Y.; Zheng, R. Sentence Similarity based on Dynamic Time Warping. In Proceedings of the International Conference on Semantic Computing (ICSC 2007), Irvine, CA, USA, 17–19 September 2007; pp. 250–256. [CrossRef]

72. Bird, S.; Klein, E.; Loper, E. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2009.
73. Pvs, A.; Gali, K. 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, Hyderabad, India, 13–14 January 2007; pp. 21–24.
74. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; pp. 6000–6010.
75. Wang, B.; Zhao, D.; Lioma, C.; Li, Q.; Zhang, P.; Simonsen, J.G. Encoding word order in complex embeddings. In Proceedings of the International Conference on Learning Representations, Online, 26 April–1 May 2020.
76. Ramesh, G.; Doddapaneni, S.; Bheemaraj, A.; Jobanputra, M.; AK, R.; Sharma, A.; Sahoo, S.; Diddee, H.; J, M.; Kakwani, D.; et al. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 145–162. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.