Computer & Robotics

Enhanced Self-Attention Model for Cross-Lingual Semantic Textual Similarity in SOV and SVO Languages: Persian and English Case Study

Ebrahim Ganjalipour [a], Amir Hossein Refahi Sheikhani [a, *], Sohrab Kordrostami [a], Ali Asghar Hosseinzadeh[a]

[a] *Department of Applied Mathematics and Computer Science, Lahijan Branch, Islamic Azad University, Lahijan, Iran*

Received 12 September 2023; Accepted 21 September 2023

Abstract

Semantic Textual Similarity (STS) is considered one of the subfields of natural language processing that has gained extensive research attention in recent years. Measuring the semantic similarity between words, phrases, paragraphs, and documents plays a significant role in natural language processing and computational linguistics. Semantic Textual Similarity finds applications in plagiarism detection, machine translation, information retrieval, and similar areas. STS aims to develop computational methods that can capture the nuanced degrees of resemblance in meaning between words, phrases, sentences, paragraphs, or even entire documents which is a challenging task for languages with low digital resources. This task becomes intricate in languages with pronoun-dropping and Subject-Object-Verb (SOV) word order specifications, such as Persian, due to their distinctive syntactic structures. One of the most important aspects of linguistic diversity lies in word order variation within languages. Some languages adhere to Subject-Object-Verb (SOV) word order, while others follow Subject-Verb-Object (SVO) patterns. These structural disparities, compounded by factors like pronoun-dropping, render the task of measuring cross-lingual STS in such languages exceptionally intricate. In the context of low-resource languages like Persian, this study proposes a customized model based on linguistic properties. Leveraging pronoun-dropping and SOV word order specifications of Persian, we introduce an innovative enhancement: a novel weighted relative positional encoding integrated into the self-attention mechanism. Moreover, we enrich context representations by infusing co-occurrence information through pointwise mutual information (PMI) factors. This paper introduces a cross-lingual model for semantic similarity analysis between Persian and English texts, utilizing parallel corpora. The experiments show that our proposed model achieves better performance than other models. Ablation study also shows that our system can converge faster and is less prone to overfitting. The proposed model is evaluated on Persian-English and Persian-Persian STS-Benchmarks and achieved 88.29% and 91.65% Pearson correlation coefficients on monolingual and cross-lingual STS-B, respectively.

Keywords: Semantic Textual Similarity, English-Persian Semantic Similarity, Transformer, SOV Word Order Language, Pointwise Mutual Information

1.Introduction

In the realm of Natural Language Processing (NLP), the quest for understanding and quantifying the meaning encoded within textual content has been a central focus. A fundamental challenge in this pursuit is to measure the semantic similarity between pieces of text, which has paved the way for the emergence of the field of Semantic Textual Similarity (STS) analysis. STS aims to develop computational methods that can capture the nuanced degrees of resemblance in meaning between words, phrases, sentences, paragraphs, or even entire documents. This field holds substantial significance due to its implications across various NLP applications, such as information retrieval, question answering, machine translation, text summarization, and more. STS techniques play a pivotal role in tackling the complexities inherent in understanding natural language. Unlike traditional approaches that

* Corresponding Author. Email:ah_refahi@yahoo.com

often rely on syntactic or surface-level analysis, new STS methods delve deeper into the semantic underpinnings of language, striving to replicate the human capacity to assess and compare the meaning of textual content. As a result, STS holds the potential to enhance the capabilities of NLP systems by enabling them to discern not only the explicit but also the implicit connections between words and concepts.

Over the past years, the field of STS has witnessed remarkable growth, driven by advancements in machine learning, deep learning, and the availability of large-scale linguistic resources. Researchers have explored various methodologies, ranging from traditional feature-based models to sophisticated neural network architectures, all aimed at capturing the intricate nuances of semantic resemblance. A critical turning point in the field was the development of pre-trained language models like BERT [1], GPT [2] and their multilingual variants like XLM-R [3], which revolutionized the way textual information is encoded and compared. Despite the wealth of literature and studies on text similarity in English, there is a noticeable scarcity of research dedicated to text similarity analysis in Persian. This research gap not only raises questions about the applicability of existing models and methods to Persian but also underscores the importance of focusing on this specific linguistic context. Research on Persian language Semantic Textual Similarity is not very vast and the provided results do not meet the needs. Because of this lack of resources, the model of the process becomes more important for achieving better results in low-resource languages.

Unlike the English language In Persian, due to the SOV (subject-object-verb) word order structure, the subject and verb have positional distance from each other. The spatial distance between the verb and the subject should not reduce the attention to the relation between these two key parts of the sentence. The transformer-based models (like BERT [1], GPT [2], and their multilingual variants XLM-R [3]) designed for languages with SVO (subject-verb-object) structure have not given the necessary importance to this linguistic feature. We focus on SOV word order and pronoun-dropping properties of Persian and present our customized model.

In this paper, considering pronoun-dropping and subject-object-verb (SOV) word order specifications of Persian, we propose customized relative positional encoding in the self-attention mechanism and we use existing STS Benchmark datasets to train and evaluate the system. We take advantage of the XLM-R [3] model to build a pre-trained language model. We modify relative positional encoding and we inject co-occurrence information by the sentence-level graph of the PMI (point-wise mutual information) factor [4]. Through empirical evaluations and comparisons, we demonstrate the efficacy of our approach in STS across diverse languages, emphasizing its utility in subject-object-verb (SOV) and subject-verb-object (SVO) word order languages. Thus, our contribution can be summarized as follows:

- We introduce a new transformer-based model for Persian-English and Persian-Persian Semantic Textual Similarity, which uses the pre-trained XLM-R [3] model and sentence-level graph of PMI (point-wise mutual information).
- We remove the machine translation phase for measuring Persian-English cross-lingual semantic textual similarity
- We modify the Transformer encoder, considering pro-drop (from "pronoun-dropping") linguistic property and subject-object-verb word order of Persian.
- We inject co-occurrence information into context representation by modifying weighted relative position encoding, which can capture global sub-token mutual information.
- We conduct extensive experiments on benchmark Persian and English STS datasets. The proposed model in this research achieves better performances regarding other multilingual and deep-hybrid architectures.

The remainder of this article is organized as follows. Section 2 overviews related works on semantic textual similarity. Section 3 describes our customized Transformer-based approach to STS. Section 4 presents datasets and experimental results. Finally, Section 5 gives the conclusions.

2.Related Works

The computation of similarity between short texts was first reported in 2006 [5]. Since then, starting from 2012 in the International Workshop on Semantic Evaluation (SemEval), the task of semantic similarity has expanded beyond binary similarity or dissimilarity to compute the degree of similarity, typically represented by a numerical value ranging from 0 to 5, for each text pair or sentence [6]. This

workshop includes important aspects of natural language processing and artificial intelligence, with semantic similarity being one of them. In semantic similarity, the degree of similarity between two sentences is determined, usually on a scale from 0 to 5. Initial ideas for identifying semantic similarity between two sentences were based on semantic alignment between words in the sentences, ultimately leading to algebraic summations of word similarities [7]. However, most contemporary research in this field focuses on sentence-level semantic representation using deep learning techniques. Sentences are transformed into numerical vectors with different dimensions through these methods, capturing the meanings of words in vector spaces. Words that are closer to each other in this space have semantic similarities. The generation of word vectors is typically done using large text corpora. In English, due to its widespread use and availability of extensive text corpora, more research has been conducted in this area. However, for languages with more limited resources and corpora, such as Persian, research in this domain has been relatively scarce. Here we introduce some of the prominent research studies in cross-lingual and monolingual STS.

Multilingual BERT [1] is a groundbreaking transformer-based language model that has been pre-trained on a massive multilingual corpus. M-BERT has been shown to perform exceptionally well on various cross-lingual NLP tasks, including cross-lingual semantic similarity. Its ability to understand and generate contextually rich embeddings for multiple languages makes it a valuable tool for cross-lingual applications.

Another variant of BERT called DistilBert [8] leverages knowledge distillation during the pre-training phase and shows that it is possible to reduce the size of a BERT model by 40% while retaining 97% of its language understanding capabilities and being 60% faster.

XLM-R [3] is an extension of M-BERT that further improves cross-lingual modeling. It has been pre-trained on a vast amount of data from 100 languages and achieves state-of-the-art results on a wide range of cross-lingual NLP tasks, including semantic similarity. XLM-R's robustness and effectiveness in handling low-resource languages make it a standout choice for cross-lingual research.

Tang et al. in 2018 [9] presented a model for low-resource languages such as Spanish, Arabic, Indonesian, and Thai. They extended a monolingual semantic similarity model framework to a multilingual setting, demonstrating that, by employing a shared multilingual encoder, each sentence can exhibit different embeddings depending on the target language.

Brychcin in [10], introduced ideas in which multilingual semantic spaces are aligned within a common space using bilingual lexicons. They employed unsupervised methods to calculate sentence similarity solely based on semantic embeddings. They also demonstrated that enhancing common semantic spaces through word weighting can improve results. Their findings indicated a Pearson correlation coefficient of 61.8% in Arabic-English sentence pairs.

In [11] wordnet definitions in 7 different languages are used to create a semantic textual similarity testbed to evaluate cross-lingual textual semantic similarity methods. A document alignment task is created to be used between Wordnet glosses of syn-sets in 7 different languages. Unsupervised textual similarity methods Wasserstein distance, Sinkhorn distance, and cosine similarity—are compared with a supervised Siamese deep learning model. The task is modeled both as a retrieval task and an alignment task to investigate the hubness of the semantic similarity functions and they found that considering the problem as a retrieval and alignment problem has a detrimental effect on the results

Pires et al. [12] conducted studies on the quality of multilingual BERT for cross-lingual tasks. They performed various experiments on diverse datasets using the multilingual BERT model and achieved promising results. In some experiments conducted on two different languages, cross-lingual embeddings for sentence pairs in certain languages, such as English and Japanese, exhibited relatively low accuracy. This reduced accuracy can be attributed to the differences in linguistic structures between the two languages. Languages like English follow the SVO structure, where the typical sentence structure consists of subject-verb-object word order. In contrast, languages like Persian follow the SOV structure, where the typical sentence structure places the subject first, followed by the object, and finally the verb, usually at the end of the sentence.

We follow the recent trends in STS, that is, using Transformer based and pretrained language models. To the best of our knowledge, none of the previous Transformer based Persian STS studies considered SOV word order and pronoun-dropping specifications of the Persian language. We customize the self-attention mechanism. We take advantage of

the XLM-R [3] and also apply the linguistic properties of Persian in the proposed model.

3.The Proposed Method

In this section, the methodology of our proposed model for Persian-Persian and Persian-English STS, is presented. It consists of 3 main steps, including: (1) tokenization, (2) sentence-level graph generation and incorporating customized positional encoding for the Persian language into the attention layer, (3) computing similarity of embedding vectors of texts, generated by SOV encoder (Persian Customized model for Persian) and SVO encoder (pretrained XML-R model for English). The related symbols and notations are shown in Table 1.

Table 1:
Notations

Notation	Definition
S	training sentence set
X	input sentence embedding matrix
x_j	j-th element of X
E, P	English and Persian sentence embedding vector
Q, K, V	query/key/value matrix
W^q, W^k, W^V	learnable query/key/value weight matrix
z_i	self-attention output vector of i-th element
A	attention matrix
$\alpha_{i,j}$	i-th row and j-th column scalar of matrix A
$a_{i,j}$	weighted relative position encoding of i,j
b_{j-i}	j-i relative position embedding
n_h	the number of heads
n	the length of sentences
V_w, V_c	word and context vocabularies
c	context word
w_i	i-th word of sequence
PMI	point-wise mutual information factor
M	co-occurrence adjacency matrix
L	MSE loss function
N	Number of sentences of dataset

3.1.Tokenization

The proposed approach for measuring semantic similarity between two sentences in two different languages uses parallel corpora to transform them into a shared vector space using multilingual model embeddings, followed by fine-tuning for cross-lingual semantic similarity between the source and target languages. While various methods exist for generating a shared vector space for words, there has been less work in creating a shared vector space for sentences. The proposed method leverages parallel corpora, mapping sentences in two different languages with different structures into a common vector space. In this paper, the XLM-R model is utilized for tokenization which is pre-trained on multilingual and English data. We employ parallel corpora and embedding vectors of texts, generated by the SOV encoder (Persian Customized model) and SVO encoder (pretrained XML-R model), to create a shared vector space between Persian and English and enhance the semantic richness of the output vectors.

Our experimental results on tokenization of Persian Tests show that XLM-RoBERTa [3] in comparison to other multilingual tokenizers is a suitable pretrained tokenizer. For tokenization, we use the XLM-RoBERTa pretrained model which outperforms multilingual BERT (M-BERT) [13] on a variety of cross-lingual benchmarks, including +14.6% average accuracy on Cross-lingual Natural Language Inference, +13% average F1 score on Multilingual Question Answering, and +2.4% F1 score on NER. XLM-RoBERTa was trained on 2.5TB of created clean Common Crawl data in 100 languages (including Persian language). It provides strong gains over previously released multi-lingual models like Multilingual BERT on downstream tasks like classification, sequence labeling, and question answering. XLM-RoBERTa utilized the Sentence-piece method [14] for sub-word tokenization which performs particularly well on low-resource agglutinative languages such as Persian.

3.2.Sentence-Level PMI Graph Generation

After tokenization process, we construct a sentence-level heterogeneous graph from tokens. The output of the tokenization process is a sub-word tokens set with corresponding ids. To capture global tokens co-occurrence within-corpus or dataset, first, we eliminate high-frequency tokens and stop words (‘.’,’!’,’ آن’,’ از’,’ به’, …), so the PMI factor of them will be zero. we build a heterogeneous text graph G = (V, E). The text graph contains token nodes (V) representing all the tokens in the corpus vocabulary. The text graph also contains token-token edges (E) which are built based on local token co-occurrence within sliding windows in the corpus, with edge weights measured by point-wise mutual information (PMI). As explained in 3.1, subject (at the beginning of the sentence) and verb (at the end of the sentence) are usually far from each other in SOV order language. We want to include their co-occurrence in PMI factor. Therefore, when the sliding window is at the beginning of the sentence, we also measure co-

occurrence with the last words and vice versa. For example, if the length of the window is 4, for the first word, the co-occurrence with the last 3 words is also measured, and for the last word, the co-occurrence with the first three words is measured. For measuring the co-occurrence of words, we calculate Pointwise mutual information. PMI is an information-theoretic association measure between a pair of discrete outcomes x and y, defined as:

$$PMI(x, y) = log(P(x, y) / P(x)P(y)) \quad (1)$$

We assume a corpus of words w ∈ V_W and their contexts c ∈ V_C, where V_W and V_C are the word and context vocabularies. The words come from a textual corpus of words $w_1, w_2, \ldots, w_n$ and the contexts for word w_i are the words surrounding it in an L-sized window $w_{i-L}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+L}$. We denote the collection of observed words and context pairs as D. We use #(w, c) to denote the number of times the pair (w, c) appears in D. Similarly,

$$\#(w) = \sum_{c' \in V_c} \#(w, c') \quad (2)$$

and

$$\#(c) = \sum_{w' \in V_w} \#(w', c) \quad (3)$$

are the number of times w and c occurred in D, respectively. In our case, PMI(w, c) measures the association between a word w and a context c by calculating the log of the ratio between their joint probability (the frequency in which they occur together) and their marginal probabilities (the frequency in which they occur independently). PMI can be estimated empirically by considering the actual number of observations in a corpus:

$$PMI(w, c) = log(\#(w, c) \cdot |D| / \#(w) \cdot \#(c)) \quad (4)$$

The use of PMI as a measure of association in NLP was introduced by Church and Hanks [4] and widely adopted for word similarity tasks [15]. Working with the PMI matrix presents some computational challenges. The rows of PMI matrix contain many entries of word-context pairs (w, c) that were never observed in the corpus, for which PMI(w, c) = log 0 = $-\infty$. Not only is the matrix ill-defined, it is also dense, which is a major practical issue because of its huge dimensions. To solve this we set PMI(w, c) = 0 in cases #(w, c) = 0, resulting in a sparse matrix. We note the matrix is inconsistent, in the sense that observed, but uncorrelated word-context pairs have a negative matrix entry, while unobserved ones have 0 in their corresponding cell. For example, consider a pair of relatively frequent words (high P(w) and P(c)) that occur only once together. There is strong evidence that the words tend not to appear together, resulting in a negative PMI value, and hence a negative matrix entry. On the other hand, a pair of frequent words (same P(w) and P(c)) that is never observed occurring together in the corpus, will receive a value of 0. A sparse and consistent alternative from the NLP literature is to use the positive PMI (PPMI) metric, in which all negative values are replaced by 0:

$$PPMI(w, c) = max(PMI(w,c), 0) \quad (5)$$

A positive PMI value implies a high semantic correlation of words in a corpus, while a negative PMI value indicates little or no semantic correlation in the corpus. Therefore, we only add edges between word pairs with positive PMI values. When representing words, there is some intuition behind ignoring negative values: humans can easily think of positive associations (e.g. "دریا" means "sea" and "ماهی" means "fish") but find it much harder to invent negative ones ("دریا" means "sea" and "بیابان" means "desert"). This suggests that the perceived similarity of two words is more influenced by the

Figure 1: Examples to show the distance between subject and verb in Persian and English. In English sentences, the subject and the verb are close together, but in Persian, the verb comes at the end of the sentence and is far from the subject (examples 1-4). Examples 2 and 3 are the same, but the pronoun subject in example 2 is omitted without changing the meaning of the sentence.

positive context they share than by the negative context they share. It therefore makes some intuitive sense to discard the negatively associated contexts and mark them as uninformative (0) instead. Indeed, it was shown that the PPMI metric performs very well on semantic similarity tasks [16]. In particular, systematic comparisons of various word-context association metrics show that PMI, and more so PPMI, provide the best results for a wide range of word-similarity tasks [16, 17].
To utilize global word co-occurrence information, we use a fixed size sliding window on all documents for each sentence of NER Datasets to gather co-occurrence statistics. After Computing PMI in all dataset, we create adjacency matrix below for each sentence.

$$M_{ij} = \begin{cases} \text{PMI}(w_i, w_j) & w_i, w_j \text{ are words}, \text{PMI}(w_i, w_j) > 0 \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

M is the adjacency matrix of the words of each sentence and (w_i, w_j) are unique index of dataset words. Considering PMI co-occurrence factor, we realize three improvements in process of Persian language as a pro-dropping and SOV order language. First, considering global co-occurrence of pronoun subject and verb, the network is trained to learn the representation of the verb from sentences in which the pronoun is not omitted to sentences in which the pronoun is omitted. We realize that, by injecting global co-occurrence of pronoun and verb with PMI factor, our method improves the recognition of gender and singularity of verb (subject-verb agreement) when the pronoun is omitted. Second, benefiting from injected global word co-occurrence, presented model achieved high validation accuracy. Third, we considered not only the co-occurrence of words in the sentence, but also the co-occurrence in the whole dataset. Therefore, the model achieves stronger generalization ability to obtain stable performance. In Experiment section, we analyze the model generalization and effect of injecting PMI global word co-occurrence in recognition of semantic similarity and subject-verb agreement.

3.3.Persian linguistic Properties that Affect Self-Attention

Persian has a subject-object-verb (SOV) word order and it is not strongly left-branching. However, because of pro-drop specification of Persian, the verb of a sentence is often not apparent until the end of a sentence.
Subject and verb are key parts in expressing meaning of sentence and attention between them is very effective to generate better contextual representation of text. As you see in Figure 1, Unlike English language In Persian, subject and verb has positional distance with each other. If the attention layer does not accept the position of the whole sequence as input and truncates it (the subject and the verb are not in the same chunk), the output embedding vector does not contain the overall exact concept. In Figure 1, Examples 1-4 shows SOV (subject-object-verb) order of Persian compared to English as SVO (subject-verb-object) order language. Examples 2 and 3 are the same, but because of pro-dropping specification of Persian the pronoun subject in example 2 is omitted without changing the meaning of the sentence. According to these specifications of the Persian language, we presented our customized model.

3.4.Incorporating Customized Positional Encoding for Text Representation

Our customized model use vector embedding of XLM Roberta tokenization model [18] with suitable positional encoding for Persian language. In this section, we explain our approach for Persian-Persian and English-Persian textual similarity. In addition, we analyze the properties of Transformer and propose two specific improvements for computing Persian context vector.
The first is that according to Persian language specification (explained in 3.3), we choose to consider maximum available relative position, based on length of sequences in learning dataset. Relative positional encoding [19] hypothesized that precise relative position information is not useful beyond a certain distance, Whereas in Persian as a pro drop language, subject and verb have distance (verb appears at the end of the sentence) and also they are semantically key parts of a sentence for generating whole context. Therefore, we don't use clipping in our model and we consider relative position weight between all tokens.

The second improvement is related to injecting point wise mutual information between sequence elements of whole corpus. We built full connected graph between tokens of sentences in preprocessing step as explained in section 3.2 and We used PMI factor in attention layer. The weights of relation edge between tokens comes from generated adjacency matrix M of eq. (6). In the following we formulate our improvement to vanilla transformer encoder [20].

The Transformer encoder takes input matrix X $\in R^{n\times d}$, where n is the sequence length, d is the input embedding vector dimension. The input matrix comes from tokenization process of section 3.1. Then three learnable matrix Wq, W_k, W_v are used to project X into different spaces. Usually, the matrix size of the three matrix are all $R^{d\times dk}$, where d_k is a hyper-parameter. After that, the scaled dot product attention can be calculated by the following equations

$$\begin{aligned} &Q,K,V = XW_q, XW_k, XW_v, \\ &A_{t,j} = Q_t K_j^T, \\ &Z = Attn(K,Q,V) = soft\max\left(\frac{A}{\sqrt{d_k}}\right)V, \end{aligned} \tag{7}$$

where Q_t is the query vector of the t'th token, j is the token the t'th token attends. K_j is the key vector representation of the j'th token. The softmax is along the last dimension.

We modify and supply Relative positional information to the model on two levels: values and keys. This becomes apparent in the modified self-attention equations shown below. Customized relative positional information is supplied to the model as an additional component to the keys. We propose eq. (9) to propagate relative position edge weights which contains PMI global word co-occurrence information. b_{j-i} is learnable relative position, weighted by sigmoid of M_{ij} (M matrix created in preprocessing steps, explained in section 3.2). By eq. (10) we inject co-occurrence and relative position into self-attention layer as follow.

$$z_i = \sum_{j=1}^{n} \alpha_{ij} (x_j W^V + a_{ij}^V) \tag{8}$$

$$e_{ij} = \frac{x_i W^Q (x_j W^K + a_{ij}^K)^T}{\sqrt{d_z}} \tag{9}$$

$$a_{ij} = b_{j-i} sigmoid(M_{ij}) \tag{10}$$

where

$(x_1, \ldots, x_n)$ are input sequence elements and $(z_1, \ldots, z_n)$ are vectors of attention matrix *Attn* $(b_{-n+1},...,b_1,...,b_{n-1})$ are vectors of position encoding learnable weights. b_{j-i} represents *j-i* relative position embedding. For example, If the distance between two elements of the sentence is 3, b_3 will be the vector representing this relative position encoding and the weights of the vector will be updated in the learning process at 3 relative positions (for $a_{1,4}$). As you see in 8-10, Adding new position weights to key vector and multiplication with query vector implies more attention between corresponding sequence elements and also, using equation 10 multiplication of b_{j-i} and $sigmoid(M_{i,j})$ injects global word co-occurrence of *i* and *j* elements to the relative position encoding. In other word with this enhancement more co-occurrence of the sequence element causes more attention to relative position of them. The softmax operation remains unchanged from vanilla self-attention. For computing attention matrix we use

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} \exp e_{ij}} \tag{11}$$

In order to achieve efficient implementation, e_{ij} is computed by eq. (12).

$$e_{ij} = \frac{x_i W^Q (x_j W^K)^T + x_i W^Q (a_{ij}^K)^T}{\sqrt{d_z}} \tag{12}$$

Instead of using one group of W_q, W_k, W_v, using several groups will enhance the ability of self-attention. When several groups are used, it is called multi-head self-attention, the calculation can be formulated as follows,

$$Q^{(h)}, K^{(h)}, V^{(h)} = HW_q^{(h)}, HW_k^{(h)}, HW_v^{(h)}, \tag{13}$$

$$head^{(h)} = \text{Attention}(Q^{(h)}, K^{(h)}, V^{(h)}), \tag{14}$$

$$Multihead(H) = [head^{(1)};...;head^{(n_h)}]W_O, \tag{15}$$

where n_h is the number of heads, the superscript h represents the head index. [head(1); ...; head(n_h)] means concatenation in the last dimension. Usually $d_k \times n_h = d$, which means the output of [head(1); ...; head(n)] will be of size $R^{n \times d}$. W_o is a learnable parameter, which is of size $R^{d \times 1}$. The output of the multi-head attention will be further processed by the position-wise feedforward networks to generate the text representation vector.

The framework of our model is shown in Figure 2. Our model consists of a preprocessing module for computing PMI factor, a tokenization layer, a customized PMI weighted relative position encoding module and semantic textual similarity module.

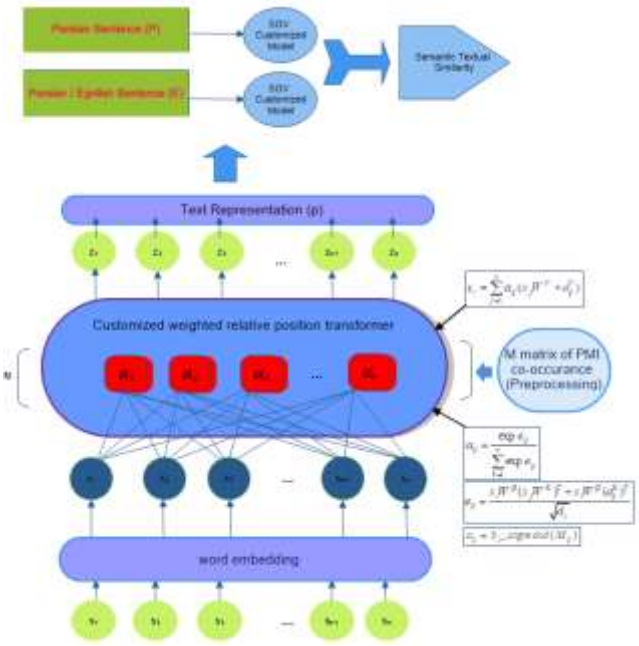


Fig.2. The framework of our model consists of a preprocessing module for computing PMI factor, a tokenization layer, a customized PMI weighted relative position encoding module and semantic textual similarity module.

3.5.Semantic Textual Similarity

After obtaining the representation vectors for each sentence using the proposed method in the architecture shown in Figure 2, the similarity between them in the vector space is computed by measuring the similarity or the inverse of the distance. Similarity metrics are distance metrics that determine the proximity or distance between two vectors. It is evident that similarity measures are inversely related to distance measures, meaning that the greater the similarity, the smaller the distance between two vectors. Various metrics are available for calculating distance, including Euclidean distance, Manhattan distance, and Minkowski distance, among others [21].

Cosine similarity is one of the most widely used metrics for measuring semantic similarity between vectors. In some articles related to semantic similarity detection, cosine similarity is transformed into angular distance. Arccos can be used for this purpose. Arccos converts cosine similarity into an angular distance that adheres to the triangle inequality. According to this approach, the absence of an angle yields better performance in detecting the semantic similarity between sentences compared to cosine similarity. Equation 16 describes how to calculate the similarity between two vectors, u and v, using Arccos [22].

$$Similarity(u, v) = -\arccos\left(\frac{u.v}{||u||||v||}\right) \quad (16)$$

Using distance-based metrics such as Euclidean and Manhattan distance, we can determine the similarity between two vectors by taking the inverse of the distance. As stated in Equation 17, the Euclidean distance calculates the shortest distance between two vectors according to the Pythagorean theorem. If x and y are two p-dimensional embedding vectors of sentences, the Euclidean and Manhattan distance between these two sentences is expressed as Equation 17 and 18.

$$D_{euc} = \sqrt{(\sum_{i=1}^{p}(x_i - y_i)^2)} \quad (17)$$

$$D_{man} = \sum_{i=1}^{p}|x_i - y_i| \quad (18)$$

3.6.Training Algorithm

Training our customized self-attention model from scratch is time consuming and resource intensive, especially in a low-resource language such as Persian, it causes overfitting. We overcome mentioned problems by using pre-training which allows models to be optimized quickly and prevents overfitting. Our model can achieve optimal performance quicker if a pre-trained model is used for generating input word embeddings. According to this we used word embedding of XLM-R as input embedding of customized model with suitable weighted positional encoding for Persian language as a SOV word order language. The proposed model

is trained in a mini-batch way, and we presented the training algorithm in revised manuscript. Embedding dimensions and the number of head are required inputs. The co-occurrence adjacency matrix is computed in preprocessing step. Tokenization and learnable parameters initialization are performed before training. During the training, the batch is sampled from STS benchmark with English and Translated Persian Pair texts and fed into customized weighted relative position Transformer encoder to get attention matrix and output embeddings. Then, the similarity module predicts the score of the STS. We use the STS score to compute the Mean Square Error (MSE) loss. Finally, the algorithm updates the model parameters according to the loss gradients.

Algorithm 1: Training algorithm for presented model.

Require: preprocessed adjacency matrix of global word co-occurrence M using (1-6)
Require: training sentences set $\boldsymbol{S}$ from STS benchmark contains sentence pairs
Require: embedding dimension d
Require: number of head n_h
Require: initialize embeddings and learnable parameters

for $t = 1, 2, 3, \dots n_{epoch}$ do
  sample a train set $\boldsymbol{S_{batch}}$ of size k
  $Loss \leftarrow 0$
  for $(\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_k})$ in $\boldsymbol{S_{batch}}$ do
    $P \leftarrow$ compute representation of Persian text (9-16)
    $E \leftarrow$ compute representation of English text (9-16)
    $score \leftarrow$ Compute similarity score of P and E
    $L(y) \leftarrow$ compute MSE loss
    $Loss \leftarrow Loss + L(y)$
    Update embeddings and learnable parameters
  end

  w.r.t the gradients using $\nabla Loss$
end

Output Similarity Score and Accuracy

3.7.Parameter Settings

We initialize the word embeddings with XLMR [3] pre-trained model. The dimensions of embeddings for sequence elements, relations, relative positions are set to 300, 128, 128, respectively. As you see parameter setting in Table 2, the hidden dimension of the self-attention layer of encoder is set at 512 including 4 heads and dimension of each head is 128. Random search strategy is used to find the optimal hyper-parameters and we use SGD with 0.9 momentum to optimize the model. The model is trained in mini-batch size of 16 and we apply dropout at a fix rate of 0.1 to avoid overfitting. In addition, we use 5-fold cross-validation. we repeated training phase 5 times separately, each time, one of the 5 subsets is used as the test set and the remaining 4 subsets are put together to form a training set.

Table 2:
Parameters setting

Parameters	Value
Batch size	16
Max sentence length	512
Learning rate	2e-5
Momentum	0.9
Number of epochs	50
Number of heads	4
Dropout rate	0.1

4.Results and Discussion

In this section, we present the results of our experiments, which demonstrate the effectiveness and robustness of our proposed model for semantic textual similarity (STS) tasks. We conducted comprehensive experiments on both monolingual and cross-lingual STS benchmarks, comparing our model against several state-of-the-art models.

4.1.Experimental Setup

4.1.1.Data Preparation

In this study, we use the Persian evaluation benchmark dataset for semantic textual similarity. We created the dataset by translating the English STS benchmark (STS-B) dataset via Google Cloud Translation API and provided various benchmark results. To assess the quality of the generated model, we need labeled data by human experts. Since there is no benchmark corpus for measuring semantic similarity between Persian and English languages, we used the English STS Benchmark corpus. For Persian-Persian benchmark, we used machine translation for all samples. For Persian-English benchmark, we translated one side of its sentence pairs into Persian using a machine translation, allowing for model evaluation. This corpus includes 8,628 sentence pairs along with a semantic similarity score ranging from 0 (lowest similarity) to 5 (highest similarity). It is divided into three parts: training (70%), validation (15%), and test (15%) datasets.

4.1.2.Evaluation Metrics

To evaluate the output of semantic similarity system, metrics such as the Pearson correlation coefficient (PCC) [23] and Spearman rank correlation [24] are applicable. The goal is to calculate the correlation between the detected similarity by the system and the actual similarity. We calculate the Pearson correlation coefficient using Equation 19:

$$\Gamma_{XY} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})}} \quad (19)$$

The above formula x_i indicates the first (or predicted) score and y_i indicate the second (or gold) score. $\bar{x}$ indicates the average of first (or predicted) scores and $\bar{y}$ indicates the average of second (or gold) scores. Predicted or gold score is used in the testing phase. If the Pearson correlation coefficient is near to one, then the obtained model is more accurate.

4.2.Comparative Experiment

In this subsection, we provide a detailed analysis of our model's performance in both monolingual and cross-lingual STS tasks. We compare our results with those of other prominent models to showcase the superiority of our approach. We have trained and tested our model on Persian-Persian and English-Persian STS-B. We divided each dataset into 5 equal subsets and used 5-fold cross-validation. we repeated the training phase 5 times separately, each time one of the 5 subsets was used as the test set, and the remaining 4 subsets were put together to form a training set. In all experiments, Pearson Correlation with Cosine similarity, Euclidean, and Manhattan distance is calculated as a metric for evaluating the performance of the model.

4.2.1.Monolingual STS (Persian-Persian)

Results of our model on Persian-Persian STS-B is given in Tables 3. On Persian-Persian STS-B, we reached 89.09% Pearson Correlation with Cosine similarity, 91.52% Pearson Correlation with Euclidean distance, and 91.65% Correlation with Manhattan distance. SOV Customized model achieved Maximum correlation with Manhattan distance better than XML-R, DistilBert, and M-BERT fine-tuned models. All compared transformer-based models obtained high cosine similarity scores, but in cases where there should not be high similarity between the sentences, they had weaker predictions and showed lower correlation with actual gold scores.

Table 3:
Results of our model in comparison to other models for Monolingual STS (Persian-Persian) on the STS Benchmark dataset

Method	Pearson Correlation with Cosine Similarity	Pearson Correlation with Euclidean distance	Pearson Correlation with Manhattan distance
M-BERT	65.06	63.66	63.65
M-BERT (Fine-tuned model)	73.77	75.34	75.37
DistilBert	66.98	67.31	67.21
DistilBert (Fine-tuned model)	72.63	74.50	75.75
XLM-R	76.57	75.31	78.37
XLM-R (Fine-tuned model)	84.57	84.11	85.68
SOV Customized	**89.09**	**91.52**	**91.65**

As shown in Figure 3 (a), when directly adopting XML-R Transformer based sentence representations to semantic textual similarity, almost all pairs of sentences achieved a similarity score between 0.6 to 1.0, even if some pairs are regarded as completely unrelated by the human annotators. In other words, the Vanilla Transformer-based sentence representations for SOV word order languages such as Persian are somehow collapsed, which means almost all sentences are mapped into a small area and therefore produce high similarity. As a result, it is inappropriate to directly apply XML-R native sentence representations for semantic matching or text retrieval. As shown in Figure 3 (b), the proposed enhancement solved this issue. The proposed customized model with weighted relative position encoding generated more accurate sentence representations and as a result, the predicted similarity was proportional to the actual similarity. Our method achieved low predicted cosine similarity in low golden similarity and higher predicted similarity for higher actual similarity.

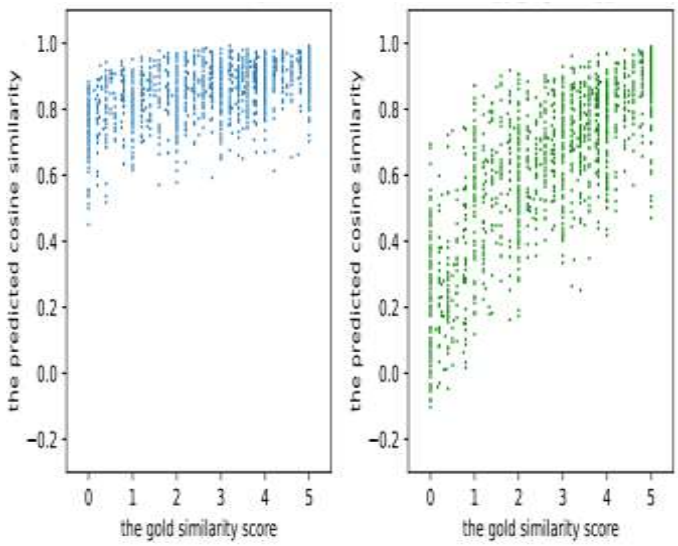


Fig.3. (a) The correlation diagram between the gold similarity score (x-axis) and the XML-R model predicted cosine similarity score (y-axis) on the Monolingual STS benchmark (Persian - Persian) dataset. (b) The correlation diagram between the gold similarity score (x-axis) and the proposed Customized model predicted the cosine similarity score (y-axis) on the Monolingual STS benchmark (Persian - Persian) dataset.

4.2.2.Cross-Lingual STS (Persian-English)

Moving to cross-lingual STS, our results on the English-Persian STS-B dataset, shown in Table 4, continue to demonstrate the effectiveness of our model. The SOV Customized model achieved substantial improvements over other models, achieving 86.98% Pearson Correlation with Cosine similarity, 87.62% Pearson Correlation with Euclidean distance, and an impressive 88.29% Pearson Correlation with Manhattan distance. These results outshine XML-R, DistilBert, and M-BERT fine-tuned models, solidifying our model's position as a state-of-the-art solution.

Table 4:
results of our model in comparison to other models for cross-lingual STS (Persian-English) on the STS Benchmark dataset

Method	Pearson Correlation with Cosine Similarity	Pearson Correlation with Euclidean distance	Pearson Correlation with Manhattan distance
M-BERT	63.88	64.03	64.11
M-BERT (Fine-tuned model)	72.61	72.39	73.19
DistilBert	65.74	65.82	66.12
DistilBert (Fine-tuned model)	69.25	69.73	70.08
XLM-R	72.02	72.85	72.28
XLM-R (Fine-tuned model)	82.39	82.35	83.47
SOV Customized	**86.98**	**87.62**	**88.29**

Our model's performance in the cross-lingual benchmark highlights its ability to effectively bridge the linguistic gap between SOV and SVO languages, such as Persian and English. These results emphasize the strength of our approach in capturing complex semantic relationships across languages. In the Persian-English cross-lingual case when we didn't utilize fine tuning the multilingual BERT model (without optimization using the parallel corpus), the correlation coefficient reaches 64.11%. However, when employing the parallel corpus, the correlation coefficient increases, and as the number of Persian-English sentence pairs in the parallel corpus increases, the Pearson correlation coefficient also rises. For instance, when we use pairs of Persian-English sentences from the parallel corpus for optimal multilingual BERT model, the correlation coefficient reaches 73.19%. Considering the linguistic features of the Persian, our model has reduced the distance between English and Persian context vectors of similar sentences. By obtaining 88.29% correlation in cross-lingual benchmark our method outperforms XML-R, DistilBert and M-BERT fined tuned models and achieved state-of-art results.

4.3.Ablation Study

To further validate the efficacy of our model, we conducted an ablation study. This study explores the impact of different components of our model on overall performance. As it is shown in Table 5, we conducted experiments on the two conditions of our model. Here, Basic RPE-Transformer indicates the Transformer model (XML-R model) which learns representation for each relative position within a clipping distance. SOV customized (suitable for SOV word order languages) indicates the self-attention model which learns representation for each relative position in the whole sentence without clipping. PMI weighted SOV Customized, indicates the complete model that includes customization for SOV word order languages and injection of PMI factor for global word co-occurrence. As can be seen from the results in Table 5, SOV customization and PMI word co-occurrence both benefit the overall results and compared to the basic model, increase accuracy by 5.97% on Persian-Persian STS-B dataset and 4.82% on Persian-English STS-B dataset, respectively.

Table 5:
Ablation study for Presented model

Model	STS-B Persian-English	STS-B Persian-Persian
Relative position encoding with clipping (Basic XML-R model)	**83.47**	**85.68**
Relative position encoding without clipping (SOV customized)	**87.91**	**89.01**
PMI weighted relative position encoding without clipping (PMI weighted SOV Customized)	**88.29**	**91.65**

When the model structure changes, the similarity scores are also different. Our method leads a significant improvement on cross-lingual and monolingual benchmarks. Reasonable explanation lies that, our method, despite the distance between subject and verb emphasizes the attention between them and injects PMI weighted global word co-occurrence between words into the word embedding. According to this, we realize that proposed model generated better context embedding and as a result obtained more STS accuracy.

4.3.1.Model Generalization

Considering Persian-English STS-B dataset, Figure 4 shows the Validation–Train Correlation curve under three conditions: basic model (the yellow curve), SOV customized (the blue curve), and PMI weighted SOV Customized (the red curve). Val PCC (Pearson correlation coefficient) and Train PCC indicate the model performance on validating set and training set respectively in the training process.

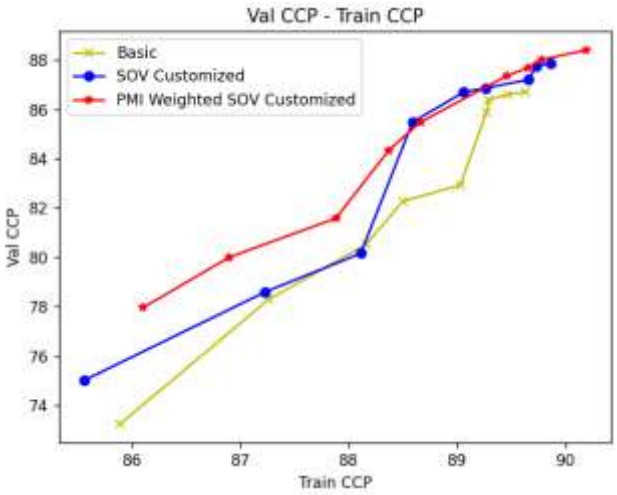
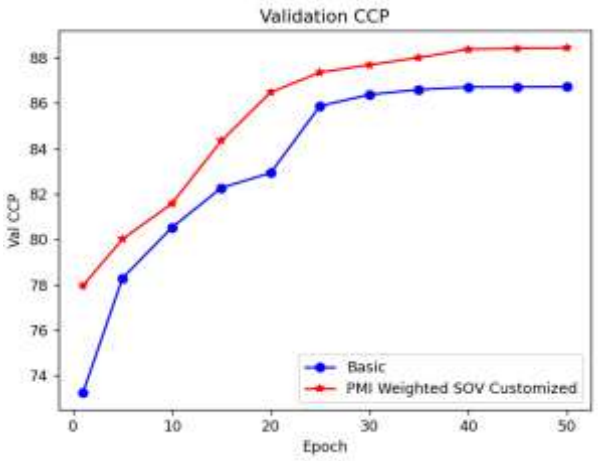


Fig.4.Validation–Train correlation curves for different model conditions.

The curves of Basic (yellow) and SOV customized (blue) in Figure 4 is close, indicating that they have similar generalization ability. The position of the red curve is on the upper side of the other curves, indicating that the Validation PCC value of complete model is higher under the same Train Correlation. Therefore, we can conclude that presented model has better generalization ability. Additionally, observing the upper right corner of Figure 4, it is obvious that Basic, SOV customized, and PMI weighted SOV Customized can reach upper and upper positions, respectively. It shows that the training level of the model is deepened in these three cases.

4.3.2.Effect of PMI Weighted Relative Position Encoding Without Clipping

Now we explore the effect of PMI weighted relative position encoding without clipping by presenting the indicators in the training process. Figure 5 shows the change of training loss and validation set correlation (accuracy) as the training epoch increases. We record the first 50 epochs to observe the situation during training. The blue curve represents the basic condition, and the red curve represents our model. Figure 5(a) shows that the training loss of our method is lower, and the convergence speed is faster during training, especially in the first 30 epochs. And the final training loss values are both close to 0.05 since they are both overfitting at that time. From Figure 5(b), it can be seen that the validation set correlation of our method increased faster, and its final value is higher. It indicates that PMI weighted relative position encoding without clipping has an inhibitory effect on overfitting.



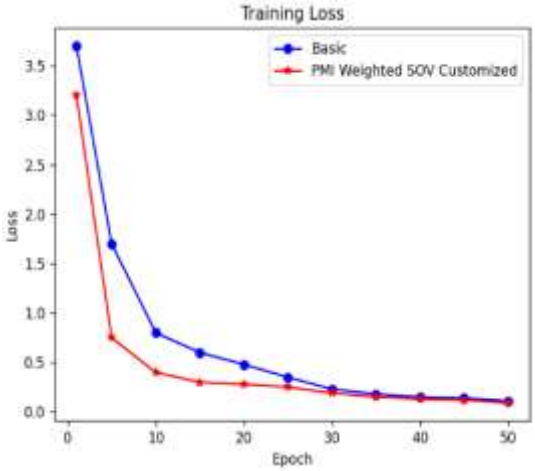


Fig.5. (a) Validation Set Correlation (b) Training loss. Indicators of training process with or without PMI weighted SOV customization.

5.Conclusions

In this paper, we presented a novel customized attention mechanism for generating context vectors of Persian texts which injected words co-occurrence information appropriated with the structure of the language. We found that, because of the linguistic properties of Persian, clipping in relative positional encoding is not suitable. We proposed fully connected relative position encoding, weighted by point-wise mutual information factor and we reached competitive performance in semantic textual similarity. An evaluation of our method on the STS-B dataset was reported. The proposed model was evaluated on Persian-English and Persian-Persian STS-Benchmarks and achieved 88.29% and 91.65% Pearson correlation coefficients on monolingual and cross-lingual STS-B, respectively.

Our future efforts will be in two directions. First, we are going to investigate a joint learning approach for semantic textual similarity on multilingual parallel STS benchmarks. Second, we want to extend our customized attention mechanism to cover a wider range of languages with distinct linguistic features and investigate how our approach works in other SOV languages such as Turkish and others.

Data Availability

STS Benchmark dataset used to support the findings of this study. Access links to the dataset is in the following Table 6.

Table 6:
Access links to datasets

Dataset	Access link
STS-B	http://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark

Ethics Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest

The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report.

Funding Statement

Author declared that no funding was received for this Research and Publication.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[3] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[4] K. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, no. 1, pp. 22-29, 1990.

[5] Y. Li, D. McLean, Z. A. Bandar, J. D. O'shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 8, pp. 1138-1150, 2006.

[6] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre, "Semeval-2012 task 6: A pilot on semantic textual similarity," in ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 2012, pp. 385-393.

[7] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 2, no. 2, pp. 1-25, 2008.

[8] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[9] X. Tang *et al.*, "Improving multilingual semantic textual similarity with shared sentence encoder for low-resource languages," *arXiv preprint arXiv:1810.08740*, 2018.

[10] T. Brychcín, "Linear transformations for cross-lingual semantic textual similarity," *Knowledge-Based Systems*, vol. 187, p. 104819, 2020.

[11] Y. Sever and G. Ercan, "Evaluating cross-lingual textual similarity on dictionary alignment problem," *Language Resources and Evaluation*, vol. 54, pp. 1059-1078, 2020.

[12] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?," *arXiv preprint arXiv:1906.01502*, 2019.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," presented at the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019.

[14] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[15] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, vol. 37, pp. 141-188, 2010.

[16] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: A computational study," *Behavior research methods*, vol. 39, no. 3, pp. 510-526, 2007.

[17] D. Kiela and S. Clark, "A systematic study of semantic vector space model parameters," presented at the Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC), 2014.

[18] Y. Liu *et al.*, "Roberta: A robustly optimized Bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[19] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[20] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] A. Singh, A. Yadav, and A. Rana, "K-means with Three different Distance Metrics," *International Journal of Computer Applications*, vol. 67, no. 10, 2013.

[22] D. Cer *et al.*, "Universal sentence encoder for English," in *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, 2018, pp. 169-174.

[23] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Noise reduction in speech processing*. Springer Science & Business Media, 2009.

[24] C. Spearman, "Correlation calculated from faulty data," *British journal of psychology*, vol. 3, no. 3, p. 271, 1910.