Received February 24, 2022, accepted March 10, 2022, date of publication March 14, 2022, date of current version March 23, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3159692

A Cross-Lingual Sentence Similarity Calculation Method With Multifeature Fusion

LINGXIN WANG, SHENGQUAN LIU, LONGYE QIAO, WEIWEI SUN, QI SUN, AND HUAQING CHENG
College of Information Science and Engineering, Xinjiang University, Ürümqi 830046, China

Corresponding author: Shengquan Liu (liu@xju.edu.cn)

This work was supported in part by the Key Projects of Scientific Research Program of Xinjiang Universities Foundation of China under Grant xjedu20191004.

ABSTRACT Cross-language sentence similarity computation is among the focuses of research in natural language processing (NLP). At present, some researchers have introduced fine-grained word and character features to help models understand sentence meanings, but they do not consider coarse-grained prior knowledge at the sentence level. Even if two cross-linguistic sentence pairs have the same meaning, the sentence representations extracted by the baseline approach may have language-specific biases. Considering the above problems, in this paper, we construct a Chinese–Uyghur cross-lingual sentence similarity dataset and propose a method to compute cross-lingual sentence similarity by fusing multiple features. The method is based on the cross-lingual pretraining model XLM-RoBERTa and assists the model in similarity calculation by introducing two coarse-grained prior knowledge features, i.e., sentence sentiment and length features. At the same time, to eliminate possible language-specific biases in the vectors, we whitened the sentence vectors of different languages to ensure that they were all represented under the standard orthogonal basis. Considering that the combination of different vectors has different effects on the final performance of the model, we introduce different vector features for comparison experiments based on the basic feature splicing method. The results show that the absolute value feature of the difference between two vectors can reflect the similarity of two sentences well. The final F1 value of our method reaches 98.97%, which is 19.81% higher than that of the baseline.

INDEX TERMS Cross-language, pre-trained model, sentence similarity, feature fusion.

I. INTRODUCTION

Sentence similarity computation is an important fundamental research topic in NLP. In recent years, sentence similarity computation methods for the same language have become increasingly mature. However, more research is needed on the study of cross-language sentence similarity. Cross-language sentence similarity refers to estimating the degree of similarity in meanings between sentences in two different languages [1]. The research results of cross-lingual sentence similarity can be applied to many applications. In cross-border e-commerce, cross-lingual sentence similarity is used to determine the most similar sentences to the target sentences from the original database [2], from which a cross-lingual intelligent question and answer customer service can be built; in machine translation, sentence similarity is used to determine the most similar target sentences to be translated from the dataset [3]; in cross-lingual text detection and text checking, sentence similarity calculation is used as the core criterion to determine the accuracy of cross-lingual text. In cross-lingual text detection and text checking, sentence similarity calculation is the core criterion that determines the accuracy of cross-lingual text detection and checking [4]; in topic tracking and detection, cross-language sentence similarity can help determine where a topic first appeared on the Internet [5]–[8]. Therefore, cross-lingual sentence similarity is an important study, and its calculation efficiency and accuracy can affect the operation efficiency of many related systems.

There are two currently accepted ideas for calculating sentence similarity: the first is to map sentence similarity to word similarity [9]. For example, "how big is Russia?" and "what's the size of Russia?" are two sentences with the same meaning, and if the NLP technique is used to split the

The associate editor coordinating the review of this manuscript and approving it for publication was Dominik Strzalka.

words, the first sentence becomes ''how/big/is/Russia/?'' and the second sentence becomes ''what's/the/size/of/Russia/?''. The second is to map sentence similarity to semantic similarity [10]–[12]. For example, ''how much does this bike cost?'' and ''how much is this bicycle?'' are two sentences that do not have any textual similarity, as one refers to ''bike'' and the other refers to ''bicycle''. However, the meanings of these two sentences are very similar. Ettinger *et al.* [13] obtained results by extracting sentence vectors of sentences that contain structural and semantic information about the sentences and then calculating the cosine similarity of the two sentence vectors. In this paper, the study of sentence similarity is based on the semantics of the text, which refers to the level of textual meaning. In deep learning, we feed sentences into the model, and the output dense vector contains the semantic information of that sentence.

Currently, most sentence similarity studies mainly focus on monolingual languages and have achieved good results, and cross-lingual sentence similarity has also achieved good results on some high-resource languages [14]. In Xinjiang, on the other hand, cross-linguistic sentence similarity research on small languages, mainly Uyghur, still needs more attention. At present, for cross-lingual sentence similarity, traditional statistical or classification methods are not effective, and to track topic hotspots in depth, the latest studies mainly consider combining topic-keyword linkage or topic-sentiment linkage. Earlier monolingual sentence similarity studies mainly used various word matching algorithms [15] and later improved model performance by introducing various word alignment methods [16]. In terms of model structure, Ranasinghe *et al.* [17] used an RNN model to compute sentence similarity, while Pontes *et al.* [18] used a combination of RNNs and CNNs to compute sentence similarity. With the development of pretrained language models such as BERT [19], through which contextual information-based word representations are extracted more accurately, the latest research uses pretrained language models to extract sentence embedding representations to compute sentence similarity [20]. However, these methods are only applicable to monolingual sentence similarity computation and are not applicable to cross-lingual sentence similarity. The approach proposed by Seki [21] is not inherently cross-linguistic; he first translated a sentence pair and then computed it using a monolingual sentence similarity model. The machine translation process generates a certain amount of error, which accumulates in the final sentence similarity calculation result, so this method has some problems. Liu *et al.* [22] mapped Chinese and other minority languages into a common semantic space for bilingual word vector pretraining, which avoids the use of machine translation models, so there is no additional error transfer. However, they did not consider introducing other prior knowledge of the sentence, so the model has no other auxiliary information to help in understanding the semantics of the text. For example, the structural features of the sentence can help in understanding the semantics, but reference [23] shows that the structural features extracted by the model are biased; thus, artificially introducing prior features would be better than the model extracting them by itself, and ultimately it would be very helpful for the model to understand the semantics of the sentence. The current mainstream approach is to use a multilingual pretraining model [24] to encode cross-lingual sentence pairs separately and then calculate their vector similarity. In this paper, we use the cross-lingual pretraining model XLM-RoBERTa [25]. XLM-RoBERTa is a cross-lingual pretraining model released by the Facebook AI team in November 2019 as an updated version of its original XLM model [26]; they are both language models based on the BERT structure, and both rely on masked language models for training. Compared to the original version, XLM-RoBERTa has two main improvements: first, changing the backbone to RoBERTa, and second, significantly increasing the amount of training data, using over 2 TB of the preprocessed CommonCrawl dataset to train cross-linguistic representations in an unsupervised manner.

We use a cross-linguistic pretraining model XLM-RoBERTa combined with a Siamese network [27] to extract the deep semantic information of the sentences and thus comprehensively represent the semantic feature information of the sentences. Siamese networks are two networks that have the same structure and share parameters [28]. In the training and validation process, we transform the problem into a binary classification problem and use the F1 value as the metric of model performance. In the inference stage, the problem can be regarded as a binary classification problem, the similarity value of sentences can be calculated using cosine similarity, and then, whether two sentences are similar can be determined. The advantage of using Siamese networks is that the input can be processed conveniently, and for the sentence pair problem, two different sentences are input to two networks with shared weights. Since the number of training samples in this paper is relatively small, overfitting is bound to occur during model training, so we mitigate model overfitting by introducing a variant of the flooding method [29]. The experimental results show that the sentence similarity calculation of this paper are more accurate than the results of Liu *et al.* [22] and Zhang *et al.* [30]

II. RELATED WORK

In recent years, many researchers in China and abroad have been investigating sentence similarity tasks, and in this section, we focus on analyzing research on existing multifeature fusion methods. Ferreira *et al.* [31] proposed a method for computing sentence similarity by fusing semantic, syntactic, and lexical features based on Word2Vec and syntactic analysis, which can solve the problem of word order and the semantics of the whole sentence. Shajalal and Aono [32] proposed the use of bilingual word-level semantics to calculate the semantic similarity between sentences, first calculating the word-level semantics of the sentences using Word2Vec and WordNet, respectively, and then combining the two semantics to calculate the final similarity, which outperformed the results calculated using these two

TABLE 1. L1 parametric values of $U - V$ vectors.

label	0	0	1	0	1	0	1	1
$\|U-V\|_1$	**897.558**	**995.825**	379.109	**761.850**	296.763	**718.165**	407.642	237.346

methods alone. Liang *et al.* [33] proposed an improved sentence similarity algorithm based on the vector space model. They argued that the traditional vector space takes a single word as the linguistic unit of a sentence and only considers the statistical information of the word without considering the semantic information of the word, so their proposed improved algorithm takes the concept as the minimum unit of the sentence, improves the accuracy of the sentence similarity algorithm through concept abstraction and specialized classification, and achieves good results in question and answer systems. Luo *et al.* [34] proposed a method for text similarity computation using syntactic and semantic information that considers weight vector dependency trees as both the semantic information and structural information of sentences. Ruan *et al.* [35] considered sentence morphological information but did not consider other semantic-related information of sentences. Liu and Liu. [36] combined multiple features, such as word overlap similarity, word order similarity, and dependency similarity, to calculate sentence similarity. Zhang *et al.* [30] captured global semantic features by introducing information at character and word granularity.

III. A CROSS-LINGUAL SENTENCE SIMILARITY CALCULATION METHOD WITH MULTIFEATURE FUSION

In this paper, the vector representation of training sentences is based on a supervised cross-language pretraining model [25] with a dataset of Chinese-Uyghur sentence pairs. Since it is very difficult to construct a dataset with labels as similarity values, this experiment transforms the problem into a binary classification problem with similar and dissimilar sentence labels of 1 and 0, respectively. In the test phases, the cosine similarity of the two sentence vectors is obtained by calculating their cosine values. Jawahar *et al.* [23] showed that the lower level of BERT learned some surface features of the sentence, such as the sentence length, the middle level learned syntactic features, such as the syntactic tree depth, and the higher level learned semantic features, such as the sentence tense and number of subjects. Therefore, the higher the level features are, the more likely the model is to forget some basic language structure information. In this paper, we introduce semantic sentiment and sentence length features to calculate sentence similarity, which represent the semantic and structural information of a sentence, respectively, with the aim of helping the model have more reference indicators when calculating sentence similarity. The approach proposed by Zhang *et al.* [30] combines two fine-grained features, words and characters, but the character features are too fine-grained for sentence-level semantic understanding and might be useful if used for word-sense similarity computation. Our approach is based on two coarse-grained features, sentiment and length, at the sentence level, which will help the model understand the global information of the sentence more than the character features.

The traditional BERT-based model will use the output vector of CLS locations as sentence features, but Sentence-BERT [37] found that averaging the vectors of all token outputs as sentence vectors works better, so this paper also defaults to the mean approach. The Siamese network outputs two sentence vectors as U, V, which are generally spliced into $[U; V]$ before being sent to the downstream task for training in most experiments. Based on the Sentence-BERT study, the various experiments in this paper show that if the absolute value of the difference between two sentence vectors is introduced, both $[|U - V|]$ and $[U; V; |U - V|]$ work better than $[U; V]$. The experiments show that the $\|U - V\|_1$ obtained for the dissimilar sentence pairs is much larger than that obtained for the similar pairs, which also indicates that the vector representation of the two sentences obtained by model learning is very accurate, as detailed in Table 1.

In fact, if the value of $\|U - V\|_1$ is very small, the difference between the values of the two vectors U, V at the corresponding positions is very small, which means that the two sentences are more similar; in contrast, if the value of $\|U - V\|_1$ is very large, the difference between the values of the two vectors U, V at the corresponding positions is very large, which means that the two sentences are less similar.

IV. METHOD

For the cross-lingual sentence similarity task, we build on the cross-lingual pretraining model XLM-RoBERTa [25] by introducing multifeature information of the sentences to help the model better understand the semantics. To mitigate the overfitting problem in the model training process, we introduce the flooding [38] method. Finally, to alleviate the vector anisotropy problem, we improve the performance of the model by introducing the whitening [39] operation for the extracted sentence vectors.

A. MULTIFEATURE ANALYSIS & MODEL ARCHITECTURE

In Section II, we mentioned that previous authors obtained better results by introducing a priori knowledge of various sentences. Here, we consider the introduction of sentiment as coarse-grained sentence information. In this paper, we use the open source sentiment analysis tool Snownlp[1] to perform quantitative sentiment analysis on sentences. Specifically, the sentence sentiment results are divided into three main

[1]https://github.com/isnowfy/snownlp

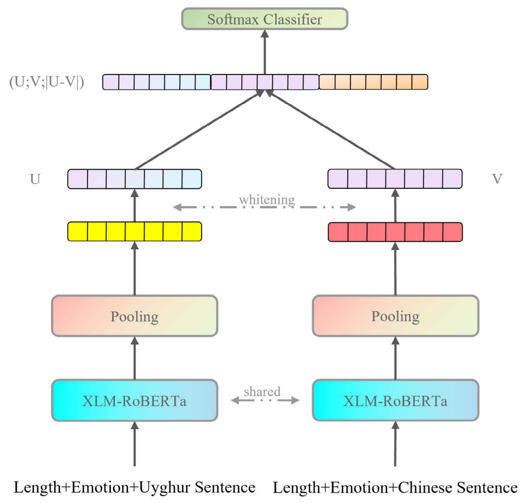


FIGURE 1. Structure of the model during training and validation with a loss function of cross-entropy loss.

categories, −1, 0, and 1, indicating negative, neutral, and positive, respectively. All sentiment states are added at the beginning of the sentence and fed into the model, and the network structure is shown in Figure 1.

We splice the three vectors U, V, and $|U - V|$ of the sentence embedding and multiply by a trainable matrix $W_t \in \mathbb{R}^{3n\times 2}$:

$$\tilde{y} = \mathrm{Softmax}(W_t(U; V; |U - V|))$$

where n is the dimension of the sentence embedding and the loss function is the cross-entropy loss function:

$$\mathcal{L} = \text{Cross-Entropy}(\tilde{y}, y)$$

In the inference stage, either the problem can be regarded as a binary classification problem or the cosine similarity can be used to calculate the similarity values of the sentences and then determine whether the two sentences are similar or not. To obtain richer information about the similarity of two sentences in the inference stage, we directly calculate the cosine of vectors U and V as the cosine similarity of two sentences. The network structure at the inference stage is shown in Figure 2.

B. FLOODING

In addition to introducing prior knowledge to help model training, we also consider a method to prevent overfitting. Ishida *et al.* [38] proposed the flooding method to cope with model overfitting by constraining the loss function so that the loss in the validation set can be reduced quadratically when overfitting occurs (as shown in Figure 3). When the training loss is less than a given threshold, we change the training target so that the validation loss does not keep rising and the overall result looks like a flooding wave. One of the flooding equations is as follows:

$$\tilde{\mathcal{L}}(\theta) = |\mathcal{L}(\theta) - b| + b, \tag{1}$$

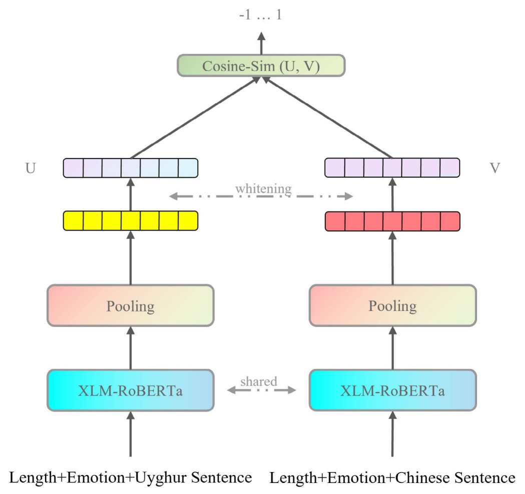


FIGURE 2. Model architecture at the inference stage.

where b is a predefined threshold value. When $\mathcal{L}(\theta) > b$ $\tilde{\mathcal{L}}(\theta) = \mathcal{L}(\theta)$, the gradient descent is performed at this time, and when $\mathcal{L}(\theta) < b$, $\tilde{\mathcal{L}}(\theta) = 2b - \mathcal{L}(\theta)$, at which point gradient ascent is performed. That is, when the training loss value is near the hyperparameter b, gradient ascent and gradient descent alternate. Assuming one step down followed by one step up and a learning rate of ε, then

$$\begin{aligned} \theta_n &= \theta_{n-1} - \varepsilon h(\theta_{n-1}) \\ \theta_{n+1} &= \theta_n + \varepsilon h(\theta_n), \end{aligned} \tag{2}$$

where $h(\theta) = \nabla_\theta \mathcal{L}(\theta)$. Bringing θ_n into θ_{n+1} gives

$$\begin{aligned} \theta_{n+1} &= \theta_{n-1} - \varepsilon h(\theta_{n-1}) + \varepsilon h\big(\theta_{n-1} - \varepsilon h(\theta_{n-1})\big) \\ &\approx \theta_{n-1} - \varepsilon h(\theta_{n-1}) + \varepsilon\big(h(\theta_{n-1}) - \varepsilon \nabla_\theta h(\theta_{n-1}) h(\theta_{n-1})\big) \\ &= \theta_{n-1} - \frac{\varepsilon^2}{2} \nabla_\theta \|h(\theta_{n-1})\|^2 \end{aligned} \tag{3}$$

The final result is equivalent to gradient descent with $\frac{\varepsilon^2}{2}$ as the learning rate and a loss function of $\|\nabla_\theta \mathcal{L}(\theta)\|^2$. Therefore, modifying the loss function by flooding is equivalent to optimizing the gradient penalty term $\|\nabla_\theta \mathcal{L}(\theta)\|^2$ after the loss function of the training set is stabilized.

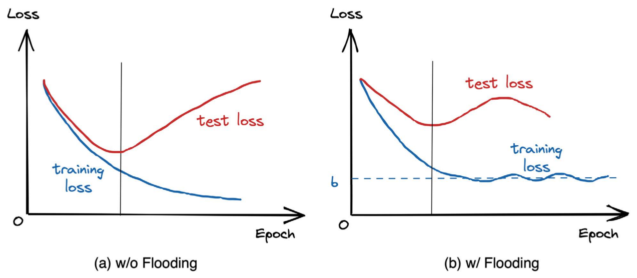


FIGURE 3. (a) shows that as the training set loss decreases, the test set loss starts to rise at a point. (b) shows that keeping the training set loss nearly constant induces the test set loss to decrease again.

The variant of flooding proposed by Wang *et al.* [29], instead of alternating gradient ascending and descending training with hyperparameter b as the timing, alternates training with different learning rates from the beginning of training.

$$\begin{aligned}\theta_n &= \theta_{n-1} - \varepsilon_1 h(\theta_{n-1}) \\ \theta_{n+1} &= \theta_n + \varepsilon_2 h(\theta_n),\end{aligned} \tag{4}$$

where $\varepsilon_1 > \varepsilon_2$. Bringing θ_n into θ_{n+1} gives

$$\begin{aligned}\theta_{n+1} &= \theta_{n-1} - \varepsilon_1 h(\theta_{n-1}) + \varepsilon_2 h(\theta_{n-1} - \varepsilon_1 h(\theta_{n-1})) \\ &\approx \theta_{n-1} - \varepsilon_1 h(\theta_{n-1}) + \varepsilon_2 (h(\theta_{n-1}) - \varepsilon_1 \nabla_\theta h(\theta_{n-1}) h(\theta_{n-1})) \\ &= \theta_{n-1} - (\varepsilon_1 - \varepsilon_2) h(\theta_{n-1}) - \frac{\varepsilon_1 \varepsilon_2}{2} \nabla_\theta \| h(\theta_{n-1}) \|^2 \\ &= \theta_{n-1} - (\varepsilon_1 - \varepsilon_2) \nabla_\theta \Bigg[\mathcal{L}(\theta_{n-1}) \\ &\quad + \frac{\varepsilon_1 \varepsilon_2}{2(\varepsilon_1 - \varepsilon_2)} \| \nabla_\theta \mathcal{L}(\theta_{n-1}) \|^2 \Bigg]\end{aligned} \tag{5}$$

The results are similar, with $(\varepsilon_1 - \varepsilon_2)$ as the learning rate and the loss function as $\mathcal{L}(\theta_{n-1}) + \frac{\varepsilon_1 \varepsilon_2}{2(\varepsilon_1 - \varepsilon_2)} \| \nabla_\theta \mathcal{L}(\theta_{n-1}) \|^2$ for training. Hyperparameter b is mainly selected by observing the training set and validation set loss curves, while the variant of flooding removes the selection problem of b, although two new parameters ε_1 and ε_2 are introduced, as long as $(\varepsilon_1 - \varepsilon_2) \in [1e-6, 1e-2]$. Figure 4 and Figure 5 show the variation of training loss and validation loss without and with flooding during our training process, respectively. The flooding parameters are set to $\varepsilon_1 = 2e-5$ and $\varepsilon_2 = 1e-6$

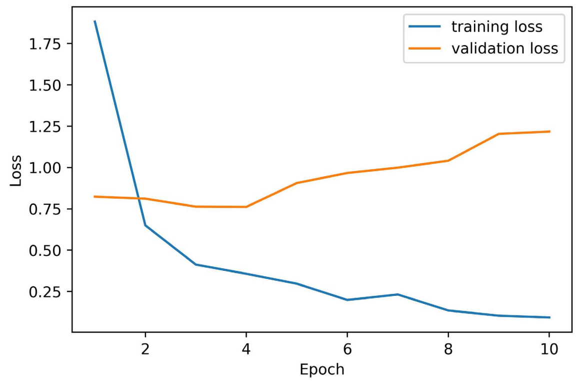


FIGURE 4. Without flooding.

C. WHITENING TRANSFORMATION

In addition to providing more features to the pretrained model to help learn semantic information, we need to perform some processing on the sentence vector to satisfy the conditions for cosine similarity calculation, which is done after the model extracts the sentence features and is therefore called postprocessing [40]. Some previous work has shown that sentence vectors extracted using models such as BERT are anisotropic, which is an important issue to be addressed when calculating sentence similarity or extracting sentence representations. Therefore, Li *et al.* [41] proposed a flow-based model to improve the quality of the obtained sentence vectors, and Su *et al.* [39] found that the whitening operation in machine

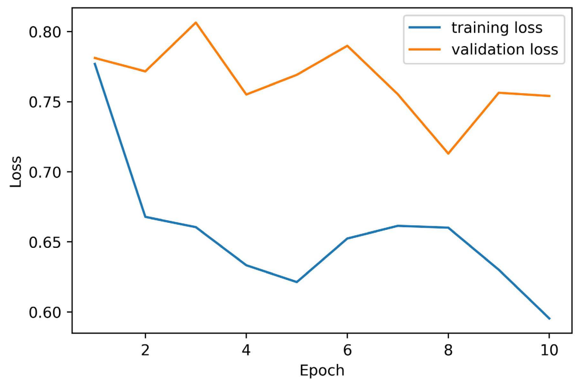


FIGURE 5. With flooding.

TABLE 2. Hyperparameter settings.

Parameters	Value
Batch size	4
Epoch	10
Optimizer	AdamW
Learning rate	1e-5
Weight decay	0.01
Max length	100
Hidden dropout	0.1
Activation function	GELU

learning can also alleviate the anisotropy problem of sentence vectors and help improve the quality of sentence representation. Whitening is a linear transformation that transforms a vector of random variables with a known covariance matrix into a new vector whose covariance is an identity matrix and has been verified to be effective in improving the text representations in bilingual word embedding mapping. Reference [42] For cross-language sentences, even if two sentences express the same meaning, the output vector of the model may contain language-specific bias, which may be caused by the different number of training samples in different languages during pretraining or some specific grammar of each language. To solve this problem, we need to jointly map the vectors of both languages to the ‘‘third party language’’ by whitening.

Sentence vector whitening is essentially a transformation of the coordinate system of a sentence vector into a standard orthogonal basis, so we can directly change the mean of the sentence vector to 0 and the covariance matrix to a unit array. We suppose the set of row vectors is $\{\boldsymbol{x}_i\}_{i=1}^N$ and perform the following transformation:

$$\tilde{\boldsymbol{x}}_i = (\boldsymbol{x}_i - \boldsymbol{\mu})\boldsymbol{W} \tag{6}$$

such that the mean of $\{\tilde{\boldsymbol{x}}_i\}$ is 0 and the covariance matrix is the unit matrix. When $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i$, the covariance matrix of the original data $\{\boldsymbol{x}_i\}$ is

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^N (\boldsymbol{x}_i - \boldsymbol{\mu})^T (\boldsymbol{x}_i - \boldsymbol{\mu}) \tag{7}$$

$$= (\frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i^T \boldsymbol{x}_i) - \boldsymbol{\mu}^T \boldsymbol{\mu} \tag{8}$$

TABLE 3. Training data samples.

Chinese sentences	Uyghur Sentences	Label
最近蔬菜价格有些上涨 (Vegetable prices have risen somewhat recently)	يېقىندىن بۇيان كۆكتات باھاسى بىر ئاز ئۆستى (Recent slight increase in vegetable prices)	1
最近蔬菜价格有些上涨 (Vegetable prices have risen somewhat recently)	يېقىندىن بۇيان كۆكتات باھاسى ئۆسمىدى (Vegetable prices have not increased recently)	0
晒太阳有很多好处 (Sunshine has many benefits)	ئاپتاپقا قاقلىنىشنىڭ نۇرغۇن پايدىسى بار (Sunshine has many benefits)	1
晒太阳有很多好处 (Sunshine has many benefits)	ئاپتاپقا قاقلىنىشنىڭ پايدىسى ئانچە چوڭ ئەمەس (The benefits of sunshine are not great)	0

Therefore, the covariance matrix of the transformed data $\{\tilde{\boldsymbol{x}}_i\}$ is:

$$\tilde{\boldsymbol{\Sigma}} = \boldsymbol{W}^T \boldsymbol{\Sigma} \boldsymbol{W} \tag{9}$$

Our ultimate goal is to solve the following equation:

$$\boldsymbol{W}^T \boldsymbol{\Sigma} \boldsymbol{W} = \boldsymbol{I} \Rightarrow \Sigma = (\boldsymbol{W}^{-1})^T \boldsymbol{W}^{-1} \tag{10}$$

Since the covariance matrix $\boldsymbol{\Sigma}$ is a semipositive definite symmetric matrix, the semipositive definite symmetric matrices all have SVD decompositions of the following form:

$$\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T, \tag{11}$$

where $\boldsymbol{U}$ is an orthogonal matrix, $\boldsymbol{\Lambda}$ is a diagonal array, and the diagonal elements are all positive, so it is straightforward to let $\boldsymbol{W}^{-1} = \sqrt{\boldsymbol{\Lambda}}\boldsymbol{U}^T$ complete the solution:

$$\boldsymbol{W} = \boldsymbol{U}\sqrt{\boldsymbol{\Lambda}^{-1}} \tag{12}$$

Finally, we can use Formula (6) to whiten the sentence representation across languages. In particular, when calculating μ and $\boldsymbol{W}$, we use all vectors in both languages for calculation.

V. EXPERIMENTS AND RESULT ANALYSIS

A. EXPERIMENTAL ENVIRONMENT AND DATA

To evaluate the effectiveness of the proposed approach, we designed cross-linguistic sentence similarity experiments under multiple conditions. In this paper, all experiments use the PyTorch framework to load pretrained models and perform training. The software environment for the experiments is the Linux operating system, the Google Colab development platform, and a Tesla K80 graphics card. More detailed parameter details are shown in Table 2.

In cross-language sentence similarity computation, there is no open source dataset for Chinese-Uyghur sentence pairs. Therefore, we analyzed how the STS-B [43] dataset was constructed. In this paper, 15,198 Tencent news headlines were collected as Chinese data in the news context and translated into Uyghur accordingly by three Uyghur researchers, two of whom performed the translation simultaneously, and the third Uyghur researcher retranslated if the translation results were different. The 15198 Uyghur sentences obtained from the translations were combined with Chinese sentences as 15198 pairs of positive samples. Then, the three Uyghur researchers modified the Uyghur sentences one by one to invert the semantics of their texts. The new Uyghur sentences were combined with Chinese sentences as 15198 pairs of negative samples, and the final dataset consisted of 30396 pairs of Chinese-Uyghur samples. Example sentences are shown in Table 3.

B. EXPERIMENTAL DESIGN AND EVALUATION METRICS

The training set and validation set ratio is 8:2, and the validation set evaluation metric is the F1 score. The F1 value is the summed average of the precision rate (P) and recall rate (R) and integrates both precision and recall metrics, as follows:

$$\begin{aligned} P &= \frac{TP}{TP + FP} \\ R &= \frac{TP}{TP + FN} \\ F1 &= \frac{2 \times P \times R}{P + R} \end{aligned} \tag{13}$$

The inference stage is judged by the cosine similarity. Let two sentence vectors $U = [u_1, u_2, u_3, \ldots, u_n]$ and $V = [v_1, v_2, v_3, \ldots, v_n]$; the cosine similarity of these two sentence vectors is calculated as follows:

$$sim = \frac{\sum_{i=1}^{n} u_i \times v_i}{\sqrt{\sum_{i=1}^{n} (u_i)^2} \times \sqrt{\sum_{i=1}^{n} (v_i)^2}} \tag{14}$$

C. EXPERIMENTAL RESULTS AND ANALYSIS

The following experiments all train 10 epochs and record the highest F1 score on the validation set. The baseline methods compared with our method are the BiLSTM-CNN-based model proposed by Liu *et al.* [22] and the LSTM-attention-based model[2] proposed by Zhang *et al.* [30]

The method proposed by Liu *et al.* is divided into three main parts. The first part is based on cross-lingual bilingual pretraining, the second part is a cross-lingual sentence feature extractor consisting of Bi-LSTM and a CNN, and the third part is a fully connected layer for cross-lingual parallel and nonparallel sentence classification. Specifically, Liu *et al.* first pretrained bilingual word embeddings by mapping them

[2]https://github.com/XuZhangp/MGF

TABLE 4. Experimental results on our datasets.

	R (Recall)	P (Precision)	F1
BiLSTM-CNN (Liu et al., 2020) [22]	75.12%	68.64%	71.73%
LSTM-attention (Zhang et al., 2020) [30]	82.42%	76.15%	79.16%

TABLE 5. XLM-Roberta final layer output vector validation set F1 score.

	$[U;V]$	$[U;V;\|U-V\|]$	$[\|U-V\|]$
No features are added	33.13%	66.16%	83.22%
Emotional features are added	39.82%	84.73%	64.97%
Sentence length features are added	31.78%	96.85%	97%
Sentence length and emotional features are added	33.58%	80.15%	**97.51%**

to a common semantic space so that semantically similar words are close in that space. Then, the trained word vectors are fed into Bi-LSTM to obtain the features before and after the words, and the CNN is used to extract the deeper semantic features of the bilingual sentence pairs. Finally, the output vector of the encoder is fed into the fully connected layer by the element product and the absolute difference of the elements, and the output probability is used as a measure of the cross-linguistic parallel utterance pairs. The model architecture proposed by Zhang *et al.* consists of an embedding layer with multigranularity information, an encoding layer with multigranularity information fusion, a matching layer and a prediction layer. First, they use a CNN to obtain the character and word embedding representations of a sentence and then send the representations to the multigrain information fusion coding layer, which uses two LSTM methods and an attention mechanism to process the character and word representations, respectively. The output representations are finally sent to the prediction layer to determine the similarity of two sentences using the sigmoid function.

The results of training 10 epochs are shown in Table 4. The batch size is 16, the learning rate is 1e-3, and the optimizer is Adam.

Our method uses the same model pretraining technique as the method of Liu *et al*. The difference is that Liu *et al* performed manual pretraining, while we use the pretrained model directly and also consider combinations of different vectors, such as the absolute difference of elements. Liu *et al.* did not consider introducing certain features of the sentence, while Zhang *et al.* considered introducing different features of the sentence. We differ from them in that we consider the anisotropy of the vectors, and to alleviate this problem, we whiten the vectors.

A comparison of Table 5 and Table 6 clearly shows that the output of the penultimate layer using XLM-Roberta is better than the output of the last layer. One reason for this is that the features used in the pretraining process are layer-1, so the features learned in layer-1 are specific to the pretraining task; however, the sentence similarity is not a pretrained task,

TABLE 6. XLM-Roberta penultimate layer output vector verification set F1 score.

	$[U;V]$	$[U;V;\|U-V\|]$	$[\|U-V\|]$
No features are added	33.01%	42.02%	84.53%
Emotional features are added	48.25%	97.69%	97.71%
Sentence length features are added	33.47%	97.71%	98%
Sentence length and emotional features are added	33.58%	**98.07%**	97.78%

TABLE 7. Result after U, V vectors are whitened.

	$[U;V]$	$[U;V;\|U-V\|]$	$[\|U-V\|]$
No features are added	60.27%	70.31%	85.04%
Emotional features are added	63.29%	96.53%	96.88%
Sentence length features are added	61%	97.77%	97.29%
Sentence length and emotional features are added	60.41%	98.33%	**98.16%**

TABLE 8. Comparison of experimental results of the flooding variant.

	$[U;V]$	$[U;V;\|U-V\|]$	$[\|U-V\|]$
without whitening	34.83%	98.83%	97.41%
with whitening	61.97%	**98.97%**	98.32%

TABLE 9. Baseline results after introduction of a priori information and whitening.

	without whitening	with whitening
No features are added	71.73%	75.29%
Emotional features are added	72.41%	76.4%
Sentence length features are added	71.74%	77.12%
Sentence length and emotional features are added	75.91%	**80.19%**

at which point the features in layer-1 are no longer the optimal choices. In the following, we analyze the results of Table 6.

First, in terms of vector splicing, $[U; V]$ is the worst, and it is difficult for the model to distinguish anything from the two sentence vectors alone. The $[U; V; |U-V|]$ and $[|U-V|]$ effects are greatly improved compared to the $[U; V]$ effect, both of which have in common that they contain $[|U-V|]$, indicating that the absolute value of the difference between the two sentence vectors can clearly reflect the degree of similarity or dissimilarity of two sentences. In particular, $[|U-V|]$ is only half of $[U; V]$ in terms of the number of parameters in this splicing method.

Then, using the penultimate layer of features, we whitened the vector, and the results are shown in Table 7.

From the results in Table 7, for some methods with poor feature splicing, the introduction of whitening greatly improves the results. The reason for this is that the vectors after whitening are in the standard orthogonal basis when the anisotropy problem of the vectors is alleviated so that each vector is more accurately represented for the sentence. This is also true for examples where fewer additional features are introduced.

TABLE 10. XLM-Roberta penultimate layer output vector verification set F1 score.

	Test sentences	Word2vec (Comparison between Chinese)	LSTM-attention (Zhang et al., 2020) [30]	Method in this article (without whitening)	Method in this article (with whitening)
1	我喜欢这个东西 (I like this stuff)/ مەن بۇ نەرسىنى ياقتۇرىمەن (This thing is liked by me)	0.71399	0.805381	0.902714	0.947112
2	一个可爱的女孩 (A lovely girl)/ قىز (A girl)	0.62112	0.583357	0.366666	0.374768
3	他无法战胜我 (He can't beat me.)/ ئۇ مېنى قانداقمۇ ئۇرسۇن؟ (How can he beat me?)	0.681012	-0.599861	0.851147	0.393781
4	小明以高分获得了第一名 (Xiao Ming won the first place with high score)/ شياۋ مىڭ ئۇتتۇرۇپ قويدى (Xiao Ming lost)	-0.140931	-0.333329	-0.271384	-0.604729
5	这个数据集不开源 (This dataset is not open source)/ بۇ ئوچۇق كودلۇق سانلىق مەلۇمات توپلىمى (This is an open source dataset)	-0.131727	0.13094	-0.331294	-0.344791

Second, from the results of feature introduction, adding both sentence length features and sentiment features has the best effect, and the final result also decreases after removing sentiment features, which indicates that sentiment features play a role. Next, using the output of the -2 layer and adding sentence length and sentiment features as the baseline, we introduce the flooding variant method to retrain the three different splicing methods (where the parameters of the flooding variant are $\varepsilon_1 = 2e-5$ and $\varepsilon_2 = 1e-6$), and the final results are shown in Table 8.

From the results in Table 8, the F1 score of the structure $[U; V; |U - V|]$ is still the highest, and except for the improved results of the feature splicing method $[U; V; |U - V|]$, the results of the other two feature splicing methods also have different degrees of improvement, which proves that the flooding variant can alleviate overfitting well and improve the performance of the model. There is a significant improvement in the results after using the whitening method.

In particular, to verify the generalizability of the two methods of introducing prior knowledge of sentences and whitening, we introduced these two methods into the model of Liu *et al.* [22] and trained them, and the results are shown in Table 9. The results increased to different degrees, indicating that the benefits of these two methods are not specific to a particular model.

After training, the best-performing model on the validation set was loaded for testing. Five sets of sentences were randomly input for testing, and Table 10 shows the results of the similarity computation of our method and the comparison method for the test sentences, where Word2vec is derived from the pretrained word vector provided by fastText proposed by Joulin *et al.* [44] The value of the result is in the range of [−1,1], and the larger the value is, the more similar the two sentences are.

TABLE 11. Classification accuracy of grammatically incorrect sentences with the model trained by adding various features.

	Accuracy
No features are added	60.00%
Emotional features are added	70.00%
Sentence length features are added	**80.00%**
Sentence length and emotional features are added	**80.00%**

To explore the effects of introducing sentence length and sentiment features in downstream task training on the model's understanding of sentence syntax, we designed a set of discrete prompt [45] experiments, as shown in Figure 6. Each sample includes one Chinese sentence and one Uyghur sentence, where Uyghur is provided by Uyghur researchers. We have a total of 20 training samples, 10 of which have grammatical errors, and the other 10 have no grammatical errors, with a training set to validation set ratio of 6:4. The preceding Chinese sentences are fixed to ask whether the following Uyghur sentences have grammatical errors, and a place to fill in the answers is left for the model to make predictions. We extracted the last column of the trained model in Table 6 for the experiment. There were 10 samples in the test set, with 5 samples with and 5 samples without grammatical errors. The results are shown in Table 11. Introduction of sentence length and sentiment features is helpful for the model to understand grammatical features.

In summary, the method provided in this paper helps the model train and understand the semantics by providing prior knowledge of the language, such as sentence length and sentiment features, during the training process and alleviates the vector anisotropy problem by the whitening method during postprocessing. In contrast, the baseline method only

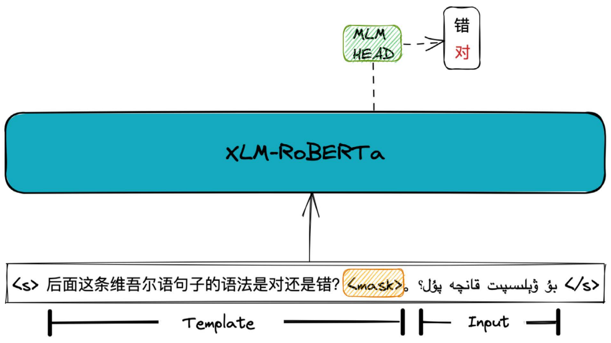

MLM HEAD
错
对
XLM-RoBERTa
<s> 后面这条维吾尔语句子的语法是对还是错? <mask>。 بۇ ۋېلسىپىت قانچە پۇل؟ </s>
Template
Input

FIGURE 6. Discrete prompt structure.

considers the words of different languages together to form a lexicon and then trains them without considering the introduction of other features, so our method works better than the baseline method.

VI. CONCLUSION

In this paper, we propose a cross-lingual sentence similarity calculation method based on semantic and structural features. By introducing the a priori features of sentiment and length in the calculation process, more accurate sentence similarity results can be obtained. The experimental results show that the pretrained model used in this paper works well on the validation set and that the final test results are better than those of the baseline method. Our next step is to expand the dataset as much as possible (the dataset will be freely available to the research community) and explore fusing additional feature information of sentences to continue improving the performance of cross-linguistic sentence similarity computation. Considering the large number of languages, nearly a hundred, even for common languages, we also studied the performance of small-sample learning [46], [47] for the many-to-many cross-language sentence similarity problem and the latest prompt-based methods [45], [46], [48], [49] to transform the problem into a pretraining task to avoid the effect of errors generated during the extraction of sentence vectors. We consider introducing adversarial training [50]–[52] and other robustness-enhancing methods to further improve the accuracy of similarity computation.

REFERENCES

[1] T. Brychcín, "Linear transformations for cross-lingual semantic textual similarity," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104819. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705119302941

[2] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1857–1863, Mar. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417414005508

[3] F. Enríquez, J. A. Troyano, and T. López-Solaz, "An approach to the use of word embeddings in an opinion classification task," *Expert Syst. Appl.*, vol. 66, pp. 1–6, Dec. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417416304833

[4] N. M. Sharef, T. Martin, K. A. Kasmiran, A. Mustapha, M. N. Sulaiman, and M. A. Azmi-Murad, "A comparative study of evolving fuzzy grammar and machine learning techniques for text categorization," *Soft Comput.*, vol. 19, no. 6, pp. 1701–1714, Jun. 2015, doi: 10.1007/s00500-014-1358-x.

[5] C. Wan, S. Jiang, C. Wang, Y. Yuan, and C. Wang, "A novel sentence embedding based topic detection method for microblogs," *IEEE Access*, vol. 8, pp. 202980–202992, 2020.

[6] H. Khalid and V. Wade, "Topic detection from conversational dialogue corpus with parallel Dirichlet allocation model and elbow method," 2020, *arXiv:2006.03353*.

[7] X. Wang and Z. Fang, "Detecting and tracking the real-time hot topics: A study on computational neuroscience," 2016, *arXiv:1608.05517*.

[8] S. Miranda, A. Znotins, S. B. Cohen, and G. Barzdins, "Multilingual clustering of streaming news," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4535–4544.

[9] M. Rafi and M. S. Shaikh, "An improved semantic similarity measure for document clustering based on topic maps," 2013, *arXiv:1303.4087*.

[10] Y. Yoo, T.-S. Heo, Y. Park, and K. Kim, "A novel hybrid methodology of measuring sentence similarity," 2021, *arXiv:2105.00648*.

[11] P. Sitikhu, K. Pahi, P. Thapa, and S. Shakya, "A comparison of semantic similarity methods for maximum human interpretability," in *Proc. Artif. Intell. Transforming Bus. Soc. (AITB)*, Nov. 2019, pp. 1–4.

[12] M. Farouk, "Measuring sentences similarity: A survey," 2019, *arXiv:1910.03940*.

[13] A. Ettinger, A. Elgohary, C. Phillips, and P. Resnik, "Assessing composition in sentence vector representations," 2018, *arXiv:1809.03992*.

[14] Y. Kim, H. Rosendahl, N. Rossenbach, J. Rosendahl, S. Khadivi, and H. Ney, "Learning bilingual sentence embeddings via autoencoding and computing similarities with a multilayer perceptron," 2019, *arXiv:1906.01942*.

[15] D. Bär, C. Biemann, I. Gurevych, and T. Zesch, "UKP: Computing semantic textual similarity by combining multiple content similarity measures," in *Proc. SEM 1st Joint Conf. Lexical Comput. Semantics Main Conf. Shared Task, 6th Int. Workshop Semantic Eval. (SemEval)*, vol. 1. Montreal, QC, Canada: Association for Computational Linguistics, 2012, pp. 435–440. [Online]. Available: https://www.aclweb.org/anthology/S12-1059

[16] M. A. Sultan, S. Bethard, and T. Sumner, "DLS@CU: Sentence similarity from word alignment and semantic vector composition," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 148–153. [Online]. Available: https://www.aclweb.org/anthology/S15-2027

[17] T. Ranasinghe, U. Research Group in Computational LinguisticsUniversity of Wolverhampton, C. Orăsan, and R. Mitkov, "Semantic textual similarity with Siamese neural networks," in *Proc. Natural Lang. Process. Deep Learn. World*, Oct. 2019, pp. 1004–1011. [Online]. Available: https://www.aclweb.org/anthology/R19-1116

[18] E. L. Pontes, S. Huet, A. Linhares, and J.-M. Torres-More, "Predicting the semantic textual similarity with Siamese CNN and LSTM," in *Proc. CORIA-TALN-RJC*, 2018, pp. 1–10.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 1–16.

[20] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 9119–9130. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.733

[21] K. Seki, "Cross-lingual text similarity exploiting neural machine translation models," *J. Inf. Sci.*, vol. 47, no. 3, pp. 404–418, Jun. 2021, doi: 10.1177/0165551520912676.

[22] C. Liu, S. Gao, Z. Yu, Y. Huang, and C. You, "Parallel sentence pair extraction method based on cross-lingual bilingual pre-training and Bi-LSTM," in *Chin. Proc. 19th Chin. Nat. Conf. Comput. Linguistics*. Haikou, China: Chinese Information Processing Society of China, Oct. 2020, pp. 457–466. [Online]. Available: https://aclanthology.org/2020.ccl-1.42

[23] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3651–3657. [Online]. Available: https://aclanthology.org/P19-1356

[24] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT sentence embedding," 2020, *arXiv:2007.01852*.

[25] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 8440–8451.

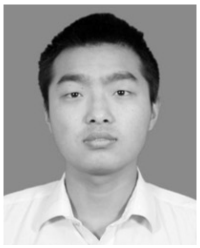
[26] G. Lample and A. Conneau, "Cross-lingual language model pretraining," in *Proc. NeurIPS*, 2019, pp. 7059–7069.

[27] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. San Diego, CA, USA, Jun. 2005, pp. 539–546.
[28] D. Chicco, *Siamese Neural Networks: An Overview*. New York, NY, USA: Springer, 2021, pp. 73–94, doi: 10.1007/978-1-0716-0826-5_3.
[29] Y. Wang, V. Peddinti, H. Xu, X. Zhang, D. Povey, and S. Khudanpur, "Backstitch: Counteracting finite-sample bias via negative steps," in *Proc. Interspeech*, Aug. 2017, pp. 1–5.
[30] X. Zhang, W. Lu, G. Zhang, F. Li, and S. Wang, "Chinese sentence semantic matching based on multi-granularity fusion model," in *Advances in Knowledge Discovery and Data Mining*, H. W. Lauw, R. C.-W. Wong, A. Ntoulas, E.-P. Lim, S.-K. Ng, and S. J. Pan, Eds. Cham, Switzerland: Springer, 2020, pp. 246–257.
[31] R. Ferreira, R. D. Lins, S. J. Simske, F. Freitas, and M. Riss, "Assessing sentence similarity through lexical, syntactic and semantic analysis," *Comput. Speech Lang.*, vol. 39, pp. 1–28, Sep. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230816000048
[32] M. Shajalal and M. Aono, "Semantic textual similarity between sentences using bilingual word semantics," *Prog. Artif. Intell.*, vol. 8, no. 2, pp. 263–272, Jun. 2019.
[33] X. Liang, D. Wang, and M. Huang, "Improved sentence similarity algorithm based on VSM and its application in question answering system," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, Oct. 2010, pp. 368–371.
[34] J. Luo, H. Shan, G. Zhang, G. Yuan, S. Zhang, F. Yan, and Z. Li, "Exploiting syntactic and semantic information for textual similarity estimation," *Math. Problems Eng.*, vol. 2021, pp. 1–12, Jan. 2021.
[35] H. Ruan, Y. Li, Q. Wang, and Y. Liu, "A research on sentence similarity for question answering system based on multi-feature fusion," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Oct. 2016, pp. 507–510.
[36] Y. Liu and Q. Liu, "Chinese sentence similarity based on multi-feature combination," in *Proc. WRI Global Congr. Intell. Syst.*, 2009, pp. 14–19.
[37] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 671–688.
[38] T. Ishida, I. Yamane, T. Sakai, G. Niu, and M. Sugiyama, "Do we need zero training loss after achieving zero training error," in *Proc. ICML*, 2020, pp. 1–16.
[39] J. Su, J. Cao, W. Liu, and Y. Ou, "Whitening sentence representations for better semantics and faster retrieval," 2021, *arXiv:2103.15316*.
[40] J. Mu, S. Bhat, and P. Viswanath, "All-but-the-top: Simple and effective postprocessing for word representations," 2017, *arXiv:1702.01417*.
[41] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," 2020, *arXiv:2011.05864*.
[42] M. Artetxe, G. Labaka, and E. Agirre, "Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 1–8, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11992
[43] J. Mu, S. Bhat, and P. Viswanath, "All-but-the-top: Simple and effective postprocessing for word representations," 2017, *arXiv:1702.01417*.
[44] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, Short Papers*, vol. 2, 2017, pp. 427–431.
[45] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, "GPT understands, too," 2021, *arXiv:2103.10385*.
[46] T. Brown *et al.*, "Language models are few-shot learners," 2020, *arXiv:2005.14165*.
[47] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, May 2021.
[48] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," 2020, *arXiv:2012.15723*.
[49] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," 2021, *arXiv:2101.00190*.
[50] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.
[51] T. Miyato, M. A. Dai, and I. Goodfellow, "Virtual adversarial training for semi-supervised text classification," 2016, *arXiv:1605.07725*.
[52] T. Miyato, S.-I. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," 2015, *arXiv:1507.00677*.

LINGXIN WANG received the B.S. degree from Wuhan City College, Wuhan, China, in 2020. He is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Xinjiang University, Ürümqi, China. His research interest includes cross-lingual sentence semantic similarity.

SHENGQUAN LIU received the M.S. degree from Beijing Jiaotong University, in 1995. He is currently a Professor with Xinjiang University. His research interests include software-defined networks and natural language processing.

LONGYE QIAO received the B.S. degree from Xinjiang University, in 2018. He is currently pursuing the Ph.D. degree with the Institute of Future Technology, Xinjiang University, Ürümqi, China. His research interests include machine learning and partial differential equations.



WEIWEI SUN received the B.S. degree from Xuhai College, China University of Mining and Technology, China, in 2019. He is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Xinjiang University, Ürümqi, China. His research interest includes natural language processing.

QI SUN received the B.S. degree from Xinjiang Agricultural University, Ürümqi, China, in 2019. She is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Xinjiang University, Ürümqi. Her research interest includes natural language processing.

HUAQING CHENG received the B.S. degree from Henan University, Kaifeng, China, in 2018. He is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Xinjiang University, Ürümqi, China. His research interest includes natural language processing.

• • •