Received November 13, 2019, accepted November 29, 2019, date of publication December 5, 2019, date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957816

Disease Prediction and Early Intervention System Based on Symptom Similarity Analysis

PEIYING ZHANG[1], XINGZHE HUANG[1], AND MAOZHEN LI[2]

[1]College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China

[2]Engineering and Design, Brunel University London, Uxbridge UB8 3PH, U.K.

Corresponding author: Peiying Zhang (25640521@qq.com)

This work was supported in part by the Fundamental Research Funds for the Central Universities of China University of Petroleum (East China) under Grant 18CX02139A, in part by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2014FQ018, and in part by the Demonstration and Verification Platform of Network Resource Management and Control Technology under Grant 05N19070040.

ABSTRACT With the development of computer technology, the electronic of medical data has become a reality. Now, how to analyze the data sufficiently to predict patient's disease and conduct early intervention has become a focused research direction. The patient's intuitive expression of feelings is also an aspect that cannot be ignored. Doctors record the pathological characteristics of patients in system. In the paper, we proposed a sentence similarity model to carry out symptom similarity analysis to achieve elementary disease prediction and early intervention, which makes use of word embedding and convolutional neural network (CNN) to extract a sentence vector that contains keyword information about the patient's feelings and symptoms. In order to increase the accuracy of sentence similarity computation, this model integrated syntactic tree and neural network into the computation process. Our main innovation is to use symptom similarity analysis model for disease prediction and early intervention. In addition, the SPO kernel is also one of the innovations. Finally, the results of experiment on Microsoft research paraphrase identification (MSRP) indicated that our model can achieve an excellent performance reached 83.9% in the terms of F1 and accuracy. Furthermore, we also conducted experiments on the data of the Semantic Textual Similarity task. Pearson correlation coefficient indicates that our result is closer to the gold standard scores, which illustrates that it can extract the key information of sentence well to realize the prediction of disease and carry out early intervention.

INDEX TERMS Convolutional neural network, disease predictions, early intervention, symptom similarity analysis.

I. INTRODUCTION

Human is expert in understanding information, while machine is expert at expressing and processing data. In this paper, we proposed a model for patient symptom similarity analysis by taking advantage of the machine's ability to process data. The model used patient's descriptions of symptoms to extract key information and achieve early prediction and intervention. Therefore, the accuracy of similarity analysis model largely determines the effectiveness of disease prediction. Nowadays, there are many researchers, who have the strong interest in sentence similarity computation and devise some models to compute the sentence similarity scores. Socher *et al.* [1] applied recurrent neural network (RNN) to find and describe images with sentences to test the accuracy of their model. However, the drawbacks of RNN are its gradients vanishing over long sequence and cannot extract dependencies between words. In order to avoid the vanishing gradients problem, the long short-term memory networks (LSTM) was proposed, which is better in learning long range dependencies. Qu *et al.* [2] and Thyagarajan *et al.* [3] used LSTM for sentence similarity measurement. The models based on LSTM have a excellent performance on tasks of semantic relatedness prediction and sentiment classification.

However, each model has pros and cons, the shortcoming of the LSTM model is needing a large amount of time to train it. Since the measurement of sentence similarity is

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

a fundamental subtask in NLP and it often used to solve larger tasks, the model would slow down the proceeding of whole task. Therefore, we should devise some novel sentence representing models to shorten the running time with an aim to solving the real issues.

For the aforementioned problem, we proposed an effective model based on CNN for the symptom similarity analysis. The main contributions of this model can be summarized as follows:

1. Since the raw symptom sentence is not accepted by model, it should be processed and encoded before it input into model. Firstly, in order to reduce the computation burden and improve the model efficiency, we proposed a kernel for extracting the main information about symptom of the sentence to preprocess the sentence. Secondly, word embedding technology is applied to map words into vectors to form the vector representation of patient's symptoms.

2. The task of our convolutional neural network is learning the semantic and syntactic features from the input tensor. Finally, we use Manhattan distance to compute the score of sentence similarity.

3. We used different datasets and evaluation criteria to evaluate the model, and obtained better results than related work, which ensures the accuracy of patient disease prediction.

The next section discusses some related work. Section 3 describes the proposed method for measuring sentence similarity. In section 4, we show and discuss the experimentation evaluation for our approach. We conclude our work in Section 5.

II. RELATED WORKS

In this paper, we proposed a disease prediction model based on symptom similarity analysis. In this model, we use sentence similarity analysis technology to analyze the similarity of patient's symptoms. Therefore, the accuracy of sentence similarity analysis determines the accuracy of the model to a large extent. Sentence similarity analysis is a basic task of natural language processing. Many researchers put forward various models to deal with sentence similarity analysis. At present, sentence similarity computation methods can be roughly categorized into four types: the models based on words match, the models based on syntactic analysis, the models based on semantic analysis and the models based on neural network. The proposed model is some different from the aforementioned models in some respects. In the following Sections, we analyze them respectively.

A. THE COMPUTATION METHODS OF SENTENCE SIMILARITY BASED ON EDIT DISTANCE

Sentence similarity computation method based on edit distance analyzes sentences from the semantic perspective. However, the disadvantage of this method is the lack of semantic understanding of sentences, which is the main reason for the low accuracy of this method. Lu *et al.* [4] proposed a method based on word2vec and editing distance, which solved the aforesaid problem in some extent. They used word2vec to obtain the word vector, and calculated the value of edit distance using the calculated vector distance as the weight of the replacement operation. The authors in [5] proposed a model based on improved chunking edit-distance to analyze the sentence similarity, which divides the sentence into NP (noun chunk), VP (verb chunk), ADJP (adjective chunk) computing the similarity between the corresponding chunks in the sentence pairs. When calculating the sentence editing distance, the higher the similarity between chunks, the lower the replacement cost. Because of the introduction of chunks, the editing distance method makes a further step in the extraction of sentence semantics. Liu *et al.* [6] proposed a model based on edit distance and dependency syntax. The model divides sentences into two layers. The first layer uses the semantic dictionary to calculate the distance. The second layer is the dominant component of the predicate central word in the sentence, whose similarity is calculated by edit distance. Finally, the two results are combined as sentence similarity. For the three methods mentioned above, in order to extract the semantic information of sentences, the weight factor is taken into account in the calculation of editing distance.

However, because editing distance only includes replace, delete and insert operations, the model extraction of sentence meaning is limited. This is one of the reasons why we introduce parsing trees in our model, which makes it possible to consider more sentence components.

B. THE COMPUTATION METHODS OF SENTENCE SIMILARITY BASED ON SYNTACTIC ANALYSIS

The sentence trunk can reflect the meaning of a sentence, therefore, the extraction of the sentence trunk requires the use of syntactic relations. Most researches focus on the dependency between adjacent words. However, in a complex sentence, the main elements of the sentence are not adjacent, which requires us to consider the syntactic relationship when analyzing sentence similarity. The authors of [7] proposed a model based on syntax and modifiers, which extracts the subject, predicate and object of sentences and preposition as the trunk of sentences to calculate the similarity. In addition, modifiers in sentences are also added to trunk, which makes the judgment result of sentences more close to the human judgment value. The authors in [8] proposed a model based on purely syntactic approach for searching similarities within sentences. In addition, position filter and counting filter are introduced into the model, which ensures the fast discarding of mismatched sequences. In the QA system, Qiu *et al.* [9] proposed a weighted syntactic analysis model, which contains 23 dependent elements, each of which is assigned different weights according to its influence on the meaning of a sentence.

Like our model, the subject, predicate and object of a sentence are considered in all three models. Compared with literature [7], the authors of [9] gave a more detailed consideration. In our model, in addition to considering syntax, we also incorporated convolutional neural network into the extraction process of sentence features.

C. THE COMPUTATION METHODS OF SENTENCE SIMILARITY BASED ON SEMANTIC ANALYSIS

Semantic characteristic [10] of sentence plays an important role in the sentence similarity computation, therefore, the scholars at home and abroad use semantic analysis to improve the accuracy of sentence similarity computation. Li *et al.* [11] presented an algorithm that takes account of semantic information and word order information implied in the sentences to further improve the accuracy of sentence similarity calculation. They constructed the word vector dynamically based entirely on the words in the compared sentences instead of high-dimensional space. The semantic similarity of two sentences is calculated using information from a structured lexical database and from corpus statistics. The use of corpora enables their models to capture common language knowledge. At the same time, the model can also be applied to different domains through corpus switching. Syntactic information is obtained through a deep parsing process that finds the phrases in each sentence. The authors of [12] leveraged semantic dependency relationship analysis to measure the degree of sentence similarity, incorporating semantic level and dependency syntactic level into the process of sentence similarity computation, and obtained satisfactory experimental results. The literature [13] presented a method for measuring the semantic similarity between texts using a corpus-base measure of word semantic similarity, and introduced a normalized and modified version of longest common subsequence (LCS) string matching algorithm. The authors in [14] presented a novel sentence similarity based on semantic. He divided the model into two subsystems, the semantic quantification and the semantic extracting function. The main function of the first subsystem is to preprocess sentences and determine the values of vectors using the WordNet semantic tree. The task of the semantic extracting subsystem is to calculate the score of sentence similarity via the semantic distance evaluated in first subsystem. The authors in [15] add offset inference algorithm to the anchored packed tree model (APT),which takes distributional composition to be a process of lexeme contextualisation, to infer the exact information in sentences. The central innovation in the compositional theory is that the APT's type structure enables the precise alignment of the semantic representation of each of the lexemes being composed. The authors shown its effectiveness for semantic composition on two benchmark phrase similarity tasks where the model achieved state-of-the-art performance. The authors in [16] proposed a method to acquire semantic feature vectors through lacarte embedding, and this method learns semantic features using corpus context. It effectively solves the problem that the set of word vectors has a common direction. Although remove stop-word and remove the top one or top few principal components are also used to solve this problem, the result is much worse than the model proposed by the authors. Finally, the authors illustrated the superiority of the model displayed in the contextual rare word dataset through Spearman coefficient between the human score and cosine similarity between word vectors. In the work [17], the authors put forward a brand new computational semantic similarity model named Direct Network Transfer (DNT). In previous work, the task of semantic similarity analysis mainly includes two types of methods. The first one is to encode each sentence into a fixed-length vector between them. The second major approach is to jointly take both sentences as input, using interaction between sentences to produce the similarity score. In DNT model, the cosine similarity of sentence embedding pairs is directly used in the loss function during transfer learning. Experiments showed that DNT model performs well in the related works of semantic similarity analysis.

Semantics is closely related to syntax and word order. Compared with the syntactic structure of a sentence, word order has a greater impact on the meaning of the sentence. In our model, the extraction of syntax is accomplished by stanford parser. In the extraction of word meaning, we drew more attention to the utilization of convolutional neural network to learn word meaning with the aim of obtaining an sentence vector containing richer sentence features [18].

D. THE COMPUTATION METHODS OF SENTENCE SIMILARITY BASED ON NEURAL NETWORK

The traditional methods of sentence similarity computation include the methods based on convolutional neural network (CNN), the method based on recurrent neural network (RNN) and so on. The authors of [19] proposed a CNN enhanced method with the aim of extracting more effective sentence features. It allows the filters to convolve not only the adjacent words in sentences but also skipped words. Sliding convolution makes it possible to extract the trunk of a sentence, although this method is not very precise. The authors in [20] proposed a novel method based on RNN to capture both textual similarity and semantic similarity between two sentences. Kim *et al.* [21] proposed a convolutional neural network model containing multiple convolutional kernels to extract sentence features through repeated convolution operations. This method, which increases the number of channels in CNN, performs well in sentence feature extraction, but it is followed by complex computation. In this model convolution kernel with different window sizes is used for multiple convolution of sentence matrix to extract matrix elements. Pontes *et al.* [22] presents a method of analyzing sentence similarity based on CNN and LSTM. This model uses the CNN network to investigate every word in the sentence to obtain the local context, and uses the LSTM network to check every word in the sentence to obtain the general context information of the sentence. Different from other neural network models, this model considers both general and local information, which makes it more sufficient to extract sentence features. The authors in [23] proposed a RNN model incorporating attention model to assess the similarity of sentence pairs. In this model, attention mechanism improves the sensitivity of the model to word similarity, which enables the model to better grasp the sensitive context information of sentences. In addition, n-gram and semantic factors are considered as surface features and applied to the extraction of sentence features. The model performed well in the 2017 SemEval cross-language semantic text similarity (STS) task.

In the aforementioned three model types, guo first transforms the sentence into a vector and then extracts the main features of the sentence by sliding convolution, which makes the extraction of the trunk of the sentence incomplete. Kim uses multiple convolution kernels to extract sentence features in an attempt to solve the shortcomings of guo's model. However, multiple convolution operations increase the computational burden of the model. Compared with the above two models, we use Stanford Parser to solve the trunk extraction problem and avoid a large amount of computation.

III. THE PROPOSED METHOD

The proposed method derives the similarity of the raw sentence from the extracted sentence trunk, which contains the subject, predicate and object of the raw sentences. We use the Stanford Parser, an open-source parser developed by the Stanford natural language processing group, to parse sentences. Primarily, the input raw sentences is judged by the syntactic analyzer whether the word sequence is legal or not, and the sentence structure is analyzed by constructing the syntactic tree to extract the subject, predicate and object of the sentence. Afterwards, the preprocessed sentence is processed by word2vec to obtain the vector representation of the sentence as the input of the CNN. In addition to extracting the features of the sentences further, the convolutional neural network can also effectively reduce the amount of computation. This is also the reason why we use CNN to extract sentence features. Finally, we use manhattan distance formula to calculate the similarity score of the sentence vector output by convolution neural network. FIGURE 1 shows the calculation process of sentence similarity score by the model.

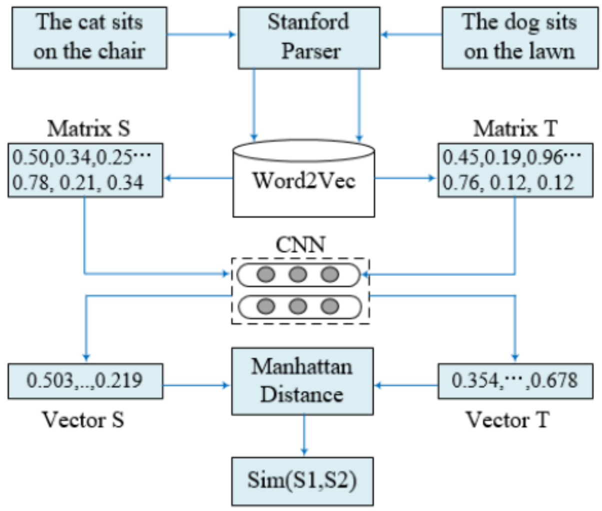


FIGURE 1. **Overview of the model architecture. Matrix S and Matrix T respectively represent the vector representation of sentence trunk, while Feature Vector S and Feature Vector T represent the sentence Feature Matrix further extracted by CNN.**

A. STANFORD PARSER

At present, most syntactic analysis methods are based on statistics, such as probabilistic context-free model, historic-based syntactic analysis model, hierarchical progressive syntactic analysis model and central word-driven syntactic analysis model.

Stanford parser [24] is an open-source parser based on probability and statistics developed by the natural language processing group at Stanford university and implemented by Java. It provides syntax tree construction and input through highly optimized context-free grammar and provides syntactic analysis functions for English, Chinese, German, Arabic, Italian, Bulgarian, Portuguese and other languages. In addition, Stanford parser performs excellent performance in word segmentation, part of speech tagging and the construction of phrase structure tree. For example, in the sentence "Syrian forces launch new attacks.", after parsing the syntax, we get the syntax tree shown in FIGURE 2. The part of speech of each word in the sentence is marked separately. The next step is to obtain the subject, predicate and object of the sentence.

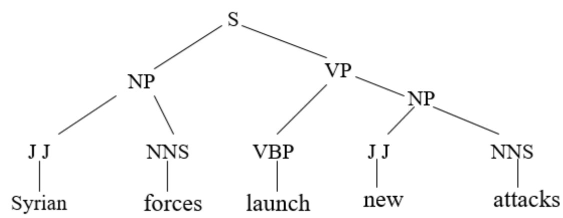


FIGURE 2. **The syntactic tree constructed by the Stanford Parser.**

B. SPO KERNEL

As mentioned before, the kernel is used to extract the subject, predicate and object (SPO) of the sentence from the syntax tree to form the trunk of the sentence. In the annotation of syntactic dependency tree generated by stanford parser, the first NN (normal noun) in NP (noun phrase) is the subject of the target sentence. VV (verb) tag in the VP (verb phrase) node is the predicate of the sentence. NN is marked as object of sentence under NP (noun phrase), PP (prepositional phrase) and ADJP (adjective phrase) tags in VP. In order to obtain the trunk of the sentence, we design a preprocessing algorithm named "Trunk Construction Algorithm", as shown in Algorithm 1. We take the syntax tree given by Stanford Parser as the input of the algorithm. The function of the algorithm is to extract the subject, predicate and object of the sentence to form a new sentence as the output. "NP" and "VP" in the algorithm represent the part of speech of words in sentences. For example, the sentence "Syrian forces launch new attacks" is simplified to "forces launch attacks" after extraction. Compared with the direct input of sentences

Algorithm 1 *Trunk Construction Algorithm*

```
Input: Syntactic Tree of Sentence
Output: Subject, Predicate and Object of Sentence
if tree.label('NP') = true then
    for s in tree.subtrees() do
        for n in s.subtrees() do
            if n.label().startswith('NN') = true then
                subject = n[0]
            end if
        end for
    end for
end if
if tree.label('VP') = true then
    for p in tree.subtrees() do
        for m in p.subtrees() do
            if m.label().startswith('VB') = true then
                predicate = m[0]
            end if
        end for
    end for
end if
if tree.label('VP') = true then
    for k in tree.subtrees() do
        for t in k.subtrees() do
            if t.label() in ['NP',' VP'] then
                for c in t.subtrees() do
                    if c.label().startswith('JJ') = true then
                        object = c[0]
                    end if
                end for
            else
                for c in t.subtrees() do
                    if c.label().startswith('JJ') = true then
                        object = c[0]
                    end if
                end for
            end if
        end for
    end for
end if
return [subject, predicate, object]
```

into the neural network, this method effectively reduces the computational burden of the model.

C. OUR CONVOLUTIONAL NEURAL NETWORK MODEL

The CNN is the core part of this paper. It is used to convolution, pooling and dimensionality reduction of input sentence matrix to obtain one-dimensional vector representation of sentences, which provides great convenience for the following sentence similarity calculation. In FIGURE 3, (c) is the structure of convolution neural network in our model, which includes twice convolution and twice pooling. In the following chapters, we introduce the model from the input layer, convolution layer, pooling layer and output layer respectively.

1) INPUT LAYER

The input layer of the model is the preprocessing process of sentences. The input sentences are parsed by Stanford Parser into sentence components including subject, predicate, object, etc. Each sentence component may contain one or more words, so we design a new component structure to describe the sentence component that contains more than one word, as shown in the Eq. (1). These components are converted into vectors by word2vec (Note that the length of the word vector we used is 50 dimensions.) and arranged in a certain sequence as the input of neural network. The specific input sequence of CNN is shown in Eq. (2).

$$W = (w_i^1, w_i^2, \ldots w_i^n), \tag{1}$$

where n represents the number of words contained in the sentence component, and w_i represents the word vector corresponding to the word.

$$Sen = (\mathbf{S}^{\mathrm{T}}, \mathbf{P}^{\mathrm{T}}, \mathbf{O}^{\mathrm{T}}), \tag{2}$$

where $\mathbf{S}^{\mathrm{T}}$ denotes for the transpose of the subject vector and $\mathbf{P}^{\mathrm{T}}$ denotes for the transpose of the predicate vector and $\mathbf{O}^{\mathrm{T}}$ denotes for the transpose of the object vector. They all have structures similar to W.

2) CONVOLUTION LAYER

As the primary part of the hidden layer, the function of convolution layer is to extract the feature of the input sentence matrix. Multiple convolution kernels are generated in the convolution layer, and these convolution kernels obtain the new sentence matrix through the convolution calculation of the sentence matrix. During the training of our model, these parameters of the convolution kernel can be adjusted through the loss value returned by the later output layer to get the parameters more suitable for calculating sentence similarity. For the trained model, the convolution kernel will scan the input sentence matrix to carry out matrix element multiplication and sum operation and superposition bias [25].

Convolution is divided into wide convolution and narrow convolution, and it is divided according to the filling mode. The essential difference between wide and narrow convolution is that the wide convolution can extract the boundary value, which is of great significance to retain the boundary value information of the sentence matrix. In addition, full padding means padding the boundary value on the periphery of the matrix to ensure that the output matrix is larger than the input matrix.

Compared with full padding, valid padding means not padding around the feature matrix, which means the output matrix is smaller than the input matrix. In the sampling process of the convolution kernel, the extraction times of the boundary value are far less than the extraction times of the intermediate value, which will dilute the sentence information contained in the boundary value of the matrix and lead to a large deviation between the similarity score and the actual value. Fulling padding and valid padding are illustrated in FIGURE 3.

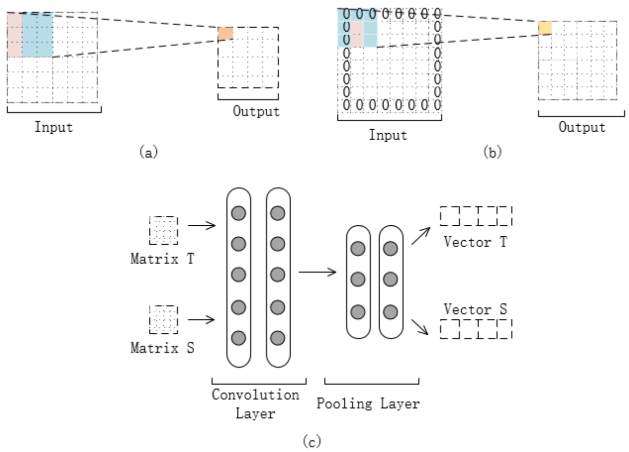


FIGURE 3. **Convolutional neural network structure.**

At the end of matrix convolution, it is necessary to use the activation function [26] to incorporate nonlinear factors into the sentence matrix. In a multilayer neural network, there is a functional relationship between the output of the upper node and the input of the lower node, which is the activation function. Due to the addition of nonlinear functions, the expression ability of neural network will be more powerful. It will no longer be a linear combination of inputs, but a model that can fit complex datasets.

In the neural network model, the commonly used activation functions include: Sigmoid function, tanh function and Relu function, which correspond to Eq. (3), Eq. (4) and Eq. (5) respectively. First, if the Sigmoid function is used for back-propagation to find the appropriate gradient, the derivative and division method will be used, resulting in a relatively large amount of computation. Secondly, when functions such as sigmoid are propagated in the back-propagation, the gradient may disappear and the neural network training cannot be completed. Based on the above considerations, Relu function is adopted as the activation function. In the training process, Relu function makes the output of some neurons equal to 0, which leads to the network sparsity and reduces the interdependence between the parameters of the sentence matrix, thus avoiding the over-fitting [27] phenomenon to some extent.

$$Sigmoid(x) = \frac{1}{1+e^{-x}}, \tag{3}$$

$$Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{4}$$

$$ReLu(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0. \end{cases} \tag{5}$$

3) POOLING LAYER

After the convolution layer, the pooling layer is used for feature dimensionality reduction, data compression and reduction of fitting degree of the sentence matrix. In addition, it can also improve the fault tolerance rate of the sentence similarity model. The specified compression of the sentence feature matrix after the convolution layer can make the sentence matrix smaller, which not only simplifies the computational complexity of the convolutional neural network, but also extracts the main features of the sentence. The main pooling methods are: Max Pooling, Maximum Pooling, Verage Pooling, K-max Pooling [28]. The average pooling considers all the components in the sentence matrix and weakens the strong eigenvalues, which may cause the nonlinear eigenvalues after activation function to counteract each other, and is not conducive to extracting the main features of the sentence. The maximum pooling directly takes the maximum eigenvalue in the target matrix, which may lead to over-fitting and poor generalization ability. K-max pooling has better compatibility by selecting k highest values in the sentence matrix to generate the subsequence of the sentence matrix, and the order of these values in the subsequence is the same as their order in the sentence matrix. It can screen out the k most active eigenvalues in the sentence matrix. Owing to the length of the two input sentences may be different, in the experiment, we adopted K-max pooling method and combined with dynamic K-max pooling operation, where k is not a fixed value, but a function of sentence length s and network depth l.

$$k = \max(k_{top}, \left\lceil \frac{L-l}{L} |s| \right\rceil). \tag{6}$$

After the sentence matrix of pooling layer, the values with strong features are left to form the final matrix representing sentences. After the dimensionality reduction of the full connection layer, the one-dimensional vector representing sentence features is obtained to calculate the similarity between two sentences.

D. SENTENCE SIMILARITY CALCULATION

For the sentence vectors obtained through the above process, we measure the similarity of sentences by calculating the distance between the vectors. At present, the calculation of similarity mainly includes manhattan distance (as shown in Eq. (7)), Euclidean distance (as shown in Eq. (8)) and cosine distance (as shown in Eq. (9)). In this paper, we use manhattan distance to calculate the similarity of sentence vectors. Because manhattan distance is not [0, 1], we need to modify the output. The detailed calculation formula as shown in Eq. (10).

$$Man(\vec{V}_x, \vec{V}_y) = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n|, \tag{7}$$

$$Euc(\vec{V}_x, \vec{V}_y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}, \tag{8}$$

$$Cos(\vec{V}_x, \vec{V}_y) = \frac{S_x \times S_y}{|S_x| \times |S_y|}, \tag{9}$$

$$score = e^{-Man(\vec{V}_x, \vec{V}_y)}, \quad score \in [0, 1] \tag{10}$$

where $\vec{V}_x, \vec{V}_y$ represent eigenvectors of sentences respectively, $\vec{V}_x$ is made up of x_i(i = 1, 2, 3 . . . n) and $\vec{V}_y$ is made up of y_i(i = 1, 2, 3 . . . n).

E. MODEL TRAINING

1) GRADIENT DESCENT ALGORITHM

Gradient descent [29] is an iterative optimization algorithm used in machine learning to find the best results. Actually, the gradient is the slope, the descent is the descent of the loss function. Learning rate is an important parameter in the process of gradient descent, that is, the rate of model learning. At the beginning of the training of the sentence similarity model, the difference of the initial parameters is too large, which leads to a higher learning rate and a larger decline step. As the point continues to decline, the learning rate decreases, leading to a decrease in the step size and then the values of loss function decreases. When the loss value decreases to a certain extent (and the loss value will not become smaller after further training), the optimal solution of the loss function is deemed to have been obtained. The loss function mentioned here is the error value between the calculated results of the similarity model and the correct results of the data set, which can be reduced by iterative algorithm. In a nutshell, the process of iterating loss function through gradient descent is the training process of the CNN model.

2) LOSS FUNCTION

In order to explore the impact of loss function [30] on training effect, we trained the model by using the minimum mean square error function and divergence loss function respectively. The minimum mean square error function is shown as follows:

$$Loss1 = \frac{1}{m}\sum_{i=1}^{m}(sim_p - sim_i)^2, \tag{11}$$

where the parameter sim_p is the predicted value of similarity score, the parameter sim_i is the similarity score calculated by the model, and the parameter m is the length of the training data set. Divergence loss function [31] is shown as follows:

$$Loss2=\frac{1}{m}\sum_{i=1}^{m}\left[q \times \log_2 \frac{q}{p}+(1-q) \times \log_2 \frac{1-q}{1-p}\right], \tag{12}$$

where p is standardized sim_p and q is standardized sim_i. Considering that the denominator p and 1-p may be zero, we apply the Laplace smoothing method to the loss function.

3) MODEL TRAINING AND PARAMETER SETTING

In the training of the model, this paper employs MSRP (Microsoft research paraphrase) training set and test set to train and test the convolutional neural network model. The sentences in the data were all taken from news sources on the internet and annotated by humans to indicate whether each pair of sentences captured the equivalent relationship between paraphrase and semantic, with a similarity score of 0 to 1. Regarding the parameters of the model, we set as follows: the pooling parameter k in the pooling layer is 3; The size of the convolution kernel is 3 times 3; The single training batch is set to 64.

At present, the training process of neural network is divided into supervised learning and unsupervised learning. In the training process of sentence similarity model, this paper adopts the supervised learning method, that is, there are two sentences to be tested and their corresponding similarity in the training data given by the data set. Through such supervised training, a convolutional neural network model with mapping ability between input and output can be obtained to calculate the similarity of sentences. In the whole process, the convolution kernel is constantly adjusted by means of gradient descent to obtain the optimal solution of the loss function. When the loss function reaches a permissible range, a sentence similarity model with high accuracy is obtained. In the training process, the step size is set as e-1, which not only guarantees the fast convergence speed to get a sentence similarity model quickly, but also avoids the over-fitting phenomenon of the model for a single data set.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Sentence similarity measurement is a highly subjective concept. Usually it can only reference to the subjective judgment of human. In order to evaluate our model, we collected the related parameters and compared it with the similar work in the experiments. In experiment 1, we use MSRP dataset to verify the performance of the model. Because the score of sentence similarity given by MSRP dataset is 0 or 1, it is more suitable for classification task. Because the output of our model is a continuous value between 0 and 1, we used STS dataset in experiment 2. STS dataset is composed of picture title, news summary, social data, etc., which proves our model's ability to process data from different fields. In experiment 3, in order to find out the shortcomings of our model, 60 pairs of sentences were selected for more detailed analysis. In addition, in order to explore the effect of sentence preprocessing on the accuracy of the model, we carried out experiment 4. Finally, experimental results show that our model is 4% more than other models in accuracy of sentence similarity calculation. The detailed data and result are as follows.

A. DATASETS AND EVALUATION METRICS

1) DATASETS

The datasets of our experiment are Microsoft research paraphrase (MSRP) released by Microsoft to make it easier for researchers to train their neural network models, which contains training data (4076 sentence pairs) and test data (1725 sentence pairs). Each pair of sentences has a similarity score given by two judges. In addition, to further illustrate the performance of the model. We conducted experiments and compared similar work on 12 datasets, all of which were presented in the Semantic Textual Similarity (STS) task. Each dataset contains many sentence pairs that cover a wide range of fields, such as news, web forum, headlines, etc. The TABLE 1 gives a detailed description of the data we used.

TABLE 1. Experimental data used in the model.

STS-2012	STS-2013	STS-2014	STS-2015
OnWN	FNWN	deft-news	answer-students
SMTeuroparl	headlines	images	images
SMTNews	OnWN	OnWN	deft-news

2) EVALUATION METRICS

In this experiment, we obtained the values of parameters in the model including true positive (TP), false positive (FP), true negative (TN) and false negative (FN). Then, these parameters are calculated to obtain the evaluation criteria of the model including F1, accuracy, recall and precision. The calculation formula of parameters is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{13}$$

$$Precision = \frac{TP}{TP + FP}, \tag{14}$$

$$Recall = \frac{TP}{TP + FN}, \tag{15}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{16}$$

where TP means that the actual result is positive and the predicted result is positive, TN means that the actual result is negative and the predicted result is negative, FP means that the actual result is negative and the predicted result is positive, FN means that the actual result is positive and the predicted result is negative.

For the STS dataset, we used the similarity score calculated by the model and the gold standard given by the offical to analyze the degree of correlation using Pearson correlation, and compared it with similar work.

B. PERFORMANCE EVALUATION

Experiment 1: We utilize MSRP datasets to train and test the model with the purpose of verifying the performance of our model. Compared with the traditional model based on convolutional neural network, we incorporated syntactic dependency tree into the model, which can extract the sentence trunk. It not only contains the main information of the sentence but also reduces the computation of the model effectively. Therefore, our model outperform than other models as show in FIGURE. 4, where the vertical axis represents different models, whereas the horizontal axis represents F1 and accuracy. We not only compare the results of different models in FIGURE. 4, but also analyze the merits and defects of each model as follows.

Hu *et al.* [32] proposed a model based on convolutional neural network for modeling sentences, which embedded words in sentence sequences and extracted the features of sentence by convolution and aggregation. Like most convolution models, they designed a large feature mapping using convolution units of shared weights and local acceptable domains to fully simulate the rich structures in word combinations.

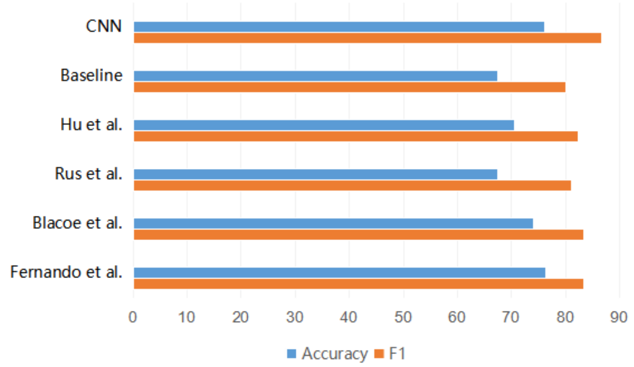


FIGURE 4. Comparison of different models.

Rus *et al.* [33] used a novel method to judge the semantic similarity between sentences, which assumes that the meaning of sentences can be captured by their syntactic components and their dependencies. They refer to syntactic components and their dependencies as feature blocks of sentences. According to the algorithm, if a suitable mapping can be found between the feature blocks of two sentences and the words have the same dependencies, we can assume that the score of sentence similarity is higher. Furthermore, sentence meaning is also affected by the word content in the feature block.

Fernando and Stevenson [34] made efficient use of word similarity information obtained from WordNet based on cognitive linguistics and proposed a model named matrix similarity method to analyze text similarity, which measures the similarity of two text segments utilizing semantic similarity. Unlike Rus, blacoe considers the semantic impact of all the words in a sentence. Blacoe used the distribution method to carry out different combinations of phrases and sentences to achieve the modeling of sentences. Although this method is very flexible to extract sentence features, it requires access to a very large corpus.

Experiment 2: In this experiment, we used Pearson's correlation between prediction scores and gold standard scores as the evaluation criteria. The gold standard contains a score between 0 and 5 for each pair of sentences, with the following interpretation: (0) The two sentences are on different topics. (1) The two sentences are not equivalent, but are on the same topic. (2) The two sentences are not equivalent, but share some details. (3) The two sentences are roughly equivalent, but some important information differs / missing. (4) The two sentences are mostly equivalent, but some unimportant details differ. (5) The two sentences are completely equivalent, as they mean the same thing. Furthermore, we have taken the experimental results of some previous models and compared them with our results, which will make our data more convincing. We selected four models for comparison with our proposed model. Among them, RNN [35] and LSTM [36] are models that use neural network to process sentence features. RNN and LSTM are good at calculating the similarity of long

TABLE 2. The comparison results: the bold number highlights one of strongest result in each dataset.

Year	Dataset	Compared methods				Our method
		RAN	LSTM	PSL	SCBOW	CNN
2012	MSRvid	0.67	**0.71**	0.60	0.45	0.57
2012	OnWN	0.63	0.56	0.63	0.64	**0.85**
2012	SMTeuroparl	0.41	0.44	0.42	0.45	**0.51**
2013	OnWN	**0.55**	0.50	0.48	0.50	0.43
2013	headlines	0.60	0.49	**0.69**	0.65	0.68
2013	FNWN	0.31	0.38	**0.38**	0.23	0.29
2014	deft-news	0.54	0.39	**0.67**	0.59	0.48
2014	images	0.58	0.51	0.65	0.64	**0.68**
2014	OnWN	0.68	0.62	0.61	0.61	**0.68**
2015	answer-forums	0.33	**0.51**	0.39	0.22	0.20
2015	answer-students	0.65	0.56	0.61	0.37	**0.67**
2015	images	0.69	0.64	0.70	0.26	**0.70**

sentences. SCBOW [37] and PSL [38] models adopt word embedding or sentence embedding technology to analyze sentence similarity. We also use word embedding technology in our model. By comparing with SCBOW and PSL, the superiority of neural network in extracting sentence similarity can be illustrated.

As we can see from TABLE 3, our model performs worse than the PSL model for the 2013 headline dataset. Firstly, we should realize that the corpus of headlines comes from news headlines. Most of these headlines are phrases, which do not have a complete grammatical structure. However, our model is based on sentence trunk to construct sentence vector. Subject, predicate and object are the main components of our sentence vector. The lack of these components leads to the degradation of our model performance. We also noted that our model outperformed other comparative models in image datasets in 2014 and 2015. The images dataset comes from the titles of some pictures on the network, such as ''the cat sits on the chair.'' The prominent feature of these sentences is that they have complete subject predicate and object and the sentences are relatively short. Stanford parser can analyze the subject predicate and object of a sentence well to construct a sentence vector. Based on the above discussion, our model is more suitable for processing sentences with complete syntactic structure.

TABLE 3. The actual values and predicted values of 60 sentence pairs.

No.	Actual Value	Predicted Value	No	Actual Value	Predicted Value
1	0.010	0.094	31	0.252	0.413
2	0.005	0.309	32	0.025	0.178
3	0.005	0.056	33	0.180	0.387
4	0.108	0.158	34	0.362	1.000
5	0.063	0.440	35	0.032	0.292
6	0.043	0.340	36	0.512	0.350
7	0.650	0.232	37	0.427	0.534
8	0.013	0.162	38	0.020	0.094
9	0.145	0.346	39	0.580	0.448
10	0.130	0.162	40	0.425	0.278
11	0.283	0.315	41	0.490	0.037
12	0.348	0.425	42	0.055	0.310
13	0.355	0.362	43	0.990	0.871
14	0.293	0.393	44	0.264	0.274
15	0.470	0.164	45	0.272	0.460
16	0.138	0.235	46	0.192	0.289
17	0.485	0.025	47	0.752	0.223
18	0.483	0.327	48	0.040	0.273
19	0.360	0.291	49	0.960	0.493
20	0.405	0.241	50	0.900	0.446
21	0.588	0.461	51	0.220	0.289
22	0.628	0.346	52	0.137	0.372
23	0.590	0.277	53	0.300	0.210
24	0.863	0.023	54	0.170	0.282
25	0.580	0.145	55	0.030	0.284
26	0.523	0.399	56	0.237	0.200
27	0.773	0.339	57	0.125	0.283
28	0.558	0.112	58	0.040	0.254
29	0.955	0.871	59	0.765	0.825
30	0.653	0.157	60	0.513	0.426

C. EXPERIMENTS ON MORE DATA

Experiment 3: To further verify whether our model can consistently perform well, we take an extra 60 pairs of sentences to evaluate the performance of our model. TABLE 3 shows the scores of human and the model on the extra 60 sentences. Note that similarity scores are in the range of [0, 1] in TABLE 3. As illustrated in TABLE 3, it can be concluded that the sentence similarity score calculated by our model is very close to that calculated by human.

D. THE INFLUENCE OF SYNTAX ON MODELS

Experiment 4: In order to further illustrate the influence of syntax on sentences, we selectively recombined the different components of sentences obtained from the syntactic tree and fed them into the convolutional neural network to analyze the results. In the experiment, we adopted the following combinations: the raw sentences (RS), the sentences after removing the stop words (RSW), the sentences after removing the adverb and the determiner (RAD), and the sentences with only the subject, predicate and object (SPO). Finally, the preprocessed sentences are used to train the model, we obtain and compare the accuracy of the model under different sentence composition conditions. The experimental results are illustrated in TABLE 4. As can be seen from TABLE 4, the SPO model shows the best performance compared with the other three models, with an accuracy of 84.6%. A raw sentence usually contains subject, predicate, object, adverbial and other sentence elements. But it is generally believed that subject predicate and object contain the most sentence information. Our model realizes the calculation of the main information of sentences. If we want to calculate the semantics more accurately, we need to take a larger dimension of word vector (note that we use 50 dimensions), and there will be new requirements for the structure of convolutional neural network. We think that an excellent model can also be obtained by setting appropriate parameters for the input sentence components, the dimension of word vector and the structure of convolutional neural network.

TABLE 4. The model accuracy is obtained by different sentence preprocessing methods.

Performance Evaluation	RS	RSW	RAD	SPO
Accuracy	0.821	0.824	0.827	0.846

E. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we will analyze and compare the experimental data to comprehensively discuss the performance of our model. To illustrate the performance of our model, we compared the experimental results of four models in experiment 1.

Obviously, we can see that the method of Hu worse than other three methods besides baseline. The reason is that Hu model only considers the influence of words on sentences without adding syntactic features. Rus preprocessed sentences with feature blocks containing sentences with syntax and dependency, which makes a small advance in F1 and accuracy. The persistence of model performance was commendably verified in experiment 2. In addition, as can be seen from experiment 3, different combinations of syntactic components have different extraction effects on sentence features. In this paper, SPO model is adopted to make F1 extraction 4 percentage points. Meanwhile, it can be learned that the experimental results of SPO model are better than other models with an accuracy of 75.8%, which shows that SPO model can capture sentence meaning better.

V. CONCLUSION

In this paper, we propose a patient symptom similarity analysis model to achieve the initial prediction and early intervention of the disease. The model uses the convolution neural network to extract the main information including symptoms and feelings of the patient in the patient description sentences to construct the sentence vector and the pooling layer to reduce the dimension of the sentence. The main innovation of this paper lies in the preprocessing of sentences and the calculation of similarity score. Firstly, we utilize SPO model to extract symptoms information as the input of neural network, which is of great significance to reduce the computational burden of the model and effectively extract the key pathological features. Secondly, we used Manhattan distance formula process the sentence vector output of the model to obtain the most similar disease prediction results. To certify our model, we develop three experiments: the first experiment shows that our model perform better than the other four models in terms of accuracy and F1. The second experiment illustrates that our model can maintain excellent performance continuously. In addition, the influence of different syntactic combinations on sentence similarity calculation is demonstrated in the third experiment.

In the future work, we will according to the domain context, extract keywords to carry out more accurate sentence similarity calculation.

ACKNOWLEDGMENT

The authors would like to thank the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. (2013). *Grounded Compositional Semantics for Finding and Describing Images With Sentences*. [Online]. Available: https://nlp.stanford.edu/

[2] Q. Xiaoyun, K. Xiaoning, Z. Chao, J. Shuai, and M. Xiuda, "Short-term prediction of wind power based on deep long short-term memory," in *Proc. IEEE PES Asia–Pacific Power Energy Eng. Conf. (APPEEC)*, Oct. 2016, pp. 1148–1152.

[3] A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proc. 30th AAAI Conf. Artif. Intell.*, Nov. 2015, pp. 2–8.

[4] L. Y, "A method of sentence similarity calculation based on word2vector and edit distance," *Comput. Knowl. Technol.*, vol. 8, no. 1, pp. 1–4, 2018.

[5] H. Zhang, Z. Yu, L. Shen, J. Guo, and X. Hong, "Naxi sentence similarity calculation based on improved chunking edit-distance," *Int. J. Wireless Mobile Comput.*, vol. 7, no. 1, pp. 48–53, 2014.

[6] Z. J. B. Liu and H. Lin, "Chinese sentence similarity computing based on improved edit-distance and dependency grammar," *Comput. Appl. Softw.*, vol. 25, no. 7, pp. 33–35, 2008.

[7] H. Deng, X. Zhu, Q. Li, and Q. Peng, "Sentence similarity calculation based on syntactic structure and modifier," *Comput. Eng.*, vol. 43, no. 9, pp. 240–244, Sep. 2017.

[8] T. P. F. Mandreoli and R. Martoglia, "A syntactic approach for searching similarities within sentences," in *Proc. 11th Int. Conf. Inf.*, 2002, pp. 1–4.

[9] C. C. G. Qiu and J. Bu, "Syntactic impact on sentence similarity measure in archive-based QA system," in *Proc. Conf. Adv. Knowl. Discovery*, 2007, pp. 2–3.

[10] J. Zwarts and Y. Winter, "Vector space semantics: A model-theoretic analysis of locative prepositions," *J. Log. Lang. Inf.*, vol. 9, no. 2, pp. 169–211, 2000.

[11] Y. Li, D. Mclean, Z. A. Bandar, D. O. James, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.

[12] B. Li, T. Liu, and B. Qin, "Chinese sentence similarity computing based on semantic dependency relationship analsis," *Appl. Res. Comput.*, vol. 20, no. 12, pp. 15–17, 2003.

[13] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Trans. Knowl. Discovery Data*, vol. 2, no. 2, pp. 1–25, 2008.

[14] C. L. Ming, "A novel sentence similarity measure for semantic-based expert systems," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 6392–6399, 2011.

[15] T. Kober, J. Weeds, J. Reffin, and D. Weir, "Improving semantic composition with offset inference," in *Proc. Comput. Res. Repository*, 2017, pp. 1–9.

[16] M. Khodak, N. Saunshi, Y. Liang, T. Ma, B. Stewart, and S. Arora, "A la carte embedding: Cheap but effective induction of semantic feature vectors," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Melbourne, SA, Australia, vol. 1, Jul. 2018, pp. 12–22.

[17] R. M. Li zhang and R. S. Wilson, "Transfer learning of sentence embedding for semantic similarity," Assoc. Comput. Linguistics, Stroudsburg, PA, USA, Tech. Rep., 2018.

[18] W. Lan and X. Wei, "Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering," in *Proc. COLING*, 2018, pp. 1–13.

[19] X. G. J. Guo and B. Yue, "An enhanced convolutional neural network model for answer selection," in *Proc. Int. World Wide Web Conf. Steering Committee*, 2017, vol. 7, no. 1, pp. 48–53.

[20] B. Ye, G. Feng, A. Cui, and M. Li, "Learning question similarity with recurrent neural networks," in *Proc. IEEE Int. Conf. Big Knowl.*, Aug. 2017, pp. 111–118.

[21] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: https://arxiv.org/abs/1408.5882

[22] E. L. Pontes, S. Huet, A. C. Linhares, and J.-M. Torres-Moreno, "Predicting the semantic textual similarity with siamese CNN and LSTM," in *Proc. Comput. Res. Repository*, 2018, pp. 311–319.

[23] W. Zhuang and E. Chang, "Neobility at SemEval-2017 task 1: An attention-based sentence similarity model," *CoRR*, vol. abs/1703.05465, 2017. [Online]. Available: http://arxiv.org/abs/1703.05465

[24] D. Chen and C. D. Manning, "A fast and accurate dependency parser using neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 740–750.

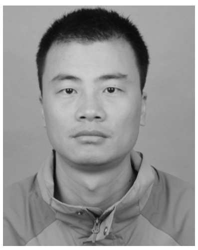
[25] I. J. Goodfellow, M. Mirza, A. Courville, and Y. Bengio, "Multi-prediction deep Boltzmann machines," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 548–556.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[27] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, vol. 1, *arXiv:1404.2188*. [Online]. Available: https://arxiv.org/abs/1404.2188

[28] G. Adam, K. Asimakis, C. Bouras, and V. Poulopoulos, *An Efficient Mechanism for Stemming and Tagging: The Case of Greek Language*. Berlin, Germany: Springer, 2010.

[29] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," in *Proc. SIAM Rev.*, 2016, pp. 223–311.
[30] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang, "Recent advances in convolutional neural networks," *Comput. Sci.*, vol. abs/1512.07108, pp. 1–38, 2015. [Online]. Available: http://arxiv.org/abs/1512.07108
[31] Z. Lu and H. Li, "A deep architecture for matching short texts," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2–6.
[32] B. Hu, Z. Lu, L. Hang, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2042–2050.
[33] V. Rus, A. C. Graesser, C. Moldovan, and N. B. Niraula, "Automatic discovery of speech act categories in educational games," in *Proc. Int. Educ. Data Mining Soc.*, 2012, p. 8.
[34] S. Fernando and M. Stevenson, "A semantic similarity approach to paraphrase detection," in *Proc. Comput. Linguistics U.K. Annu. Res. Colloq.*, 2008, pp. 45–52.
[35] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," in *Proc. ICLR*, Nov. 2015, pp. 223–311.
[36] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, no. 1, pp. 115–143, 2002.
[37] T. Kenter, A. Borisov, and M. Rijke, "Siamese CBOW: Optimizing word embeddings for sentence representations," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, Aug. 2016, pp. 941–951, doi: 10.18653/v1/P16-1089.
[38] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "From paraphrase database to compositional paraphrase model and back," *Comput. Sci.*, vol. 3, no. 2, pp. 345–358, 2015.



PEIYING ZHANG received the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, in 2019. He is currently an Associate Professor with the College of Computer Science and Technology, China University of Petroleum (East China). He has published more than 50 articles in prestigious peer-reviewed journals and conferences. His research interests include semantic computing, the future internet architecture, network virtualization, and artificial intelligence for networking.

XINGZHE HUANG is currently pursuing the master's degree with the College of Computer Science and Technology, China University of Petroleum (East China). His research interests include artificial intelligence and natural language processing.

MAOZHEN LI is currently a Professor with the Department of Electronic and Computer Engineering, Brunel University London. His main research interests include high performance computing, big data analytics, and knowledge and data engineering. He is a Fellow of both BCS and IET.