Research on Semantic Similarity of Short Text Based on Bert and Time Warping Distance

Shijie Qiu[1], Yan Niu[1], Jun Li[3,*] and Xing Li[2]

[1]*School of Computer Science, Hubei University of Technology, Wuhan, China*
[2]*China Communications Services Sci and Tech Co., Ltd., Wuhan, China*
[3]*School of Computing, Hubei University of Technology, Wuhan, 430061, China*
E-mail: 19981020@hubt.edu.cn
[*]*Corresponding Author*

Received 29 July 2021; Accepted 10 August 2021;
Publication 06 November 2021

Abstract

The research on semantic similarity of short text plays an important role in machine translation, emotion analysis, information retrieval and other AI business applications. However, according to existing short text similarity research, the characteristics of ambiguous vocabularies are difficult to be effectively analyzed, the solution of the problem caused by words order needs to be further optimized as well. This paper proposes a short text semantic similarity calculation method that combines BERT and time warping distance algorithm, in order to solve the problem of vocabulary ambiguity. The model first uses the pre trained Bert model to extract the semantic features of the short text from the whole level, and obtains a 768 dimensional short text feature vector. Then, it transforms the extracted feature vector into a point sequence in space, uses the CTW algorithm to calculate the time warping distance between the curves connected by the point sequence, and finally

Journal of Web Engineering, Vol. 20_8, 2521–2544.
doi: 10.13052/jwe1540-9589.20814

uses the weight function designed by the analysis, according to the smaller the time warpage distance is, the higher the degree of small similarity is, to calculate the similarity between short texts. The experimental results show that this model can mine the feature information of ambiguous words, and calculate the similarity of short texts with lexical ambiguity effectively. Compared with other models, it can distinguish the semantic features of ambiguous words more accurately.

Keywords: BERT, CTW, time warping distance, lexical ambiguity, semantic similarity.

1 Introduction

With the popularity and development of mobile smart terminal devices, social networks, short text data such as news summaries, microblog blog posts, and product reviews have emerged in large quantities, and mining commercially valuable information from the massive short text data has become a topic of concern for Chinese natural language processing research scholars. Short text similarity plays a huge role as the core work of AI commercial applications such as machine translation, sentiment analysis, and information retrieval.

At present, with the development of deep learning in the field of text information processing, many scholars have applied the neural network-based word vector model to the research related to text processing and achieved good results.

Shengguo Guo [1] et al. proposed a similarity calculation method combining word vector and dependent syntax, constructing a word vector model by Word2vec tool, analyzing syntactic structure by dependency analysis, and finally constructing a similarity calculation model by set weight assignment, which has a certain degree of improvement compared with direct edit distance calculation of word vector, but because the data of experimental test is mainly However, because the data of the experimental test are mainly from the corpus of Huajian it has certain limitations, and the model is mainly for the one work of English-Chinese machine translation. Xinting Liu [2] et al. proposed a sentence similarity calculation method combining word vector and frame semantics, also constructing word vector model by Word2vec and using semantic frame to analyze the overall semantics of the sentence, which has a certain degree of improvement in the calculation of Chinese semantic similarity compared with other methods. But word2vec only considers the local

information of words, and does not consider the relationship between words and words outside the local window, Jeffrey [3] et al. proposed a glove model, which mainly uses co-occurrence matrix and considers both local information and global information. However, there is still a problem in word2vec and glove models, that is, they does not consider that words have different meanings in different contexts. In these two models, the vector representation of words in different contexts is the same. So Matthew [4] et al. used a dynamic model Elmo to optimize this problem, and learned the complex usage of words through multi-layer stack LSTM. Nguyen [5] et al. proposed a short textbook based on the semantic relatedness between concepts from external knowledge sources and word embedding techniques combined with short text semantic similarity computation method by performing co-referential parsing of named entities in short text to link entities together and performing word separation to preserve the meaning of phrasal verbs and idioms, which makes the parsed features contribute to semantic similarity, but the learning model is supervised and requires feature engineering as well as a large amount of labeled data.

To address the above research shortcomings, this paper proposes a BERT [6] combined with time warping distance algorithm model for semantic similarity of short texts. BERT model is a dynamic word vector model that obtains text feature vectors containing word-word semantic and sequential structure information by pre-training a large-scale unsupervised text corpus, and its unique mask training mechanism can solve the problem that the Word2vec word vector model cannot combine the current contextual semantic dynamic expression features. The algorithm model in this paper processes the short text by the trained BERT model to generate 768-dimensional text feature vectors, then use the designed CTW algorithm to calculate the time warping distance between two short text feature vectors as a measure of text distance. Compared with other distance measurement methods, CTW performs better in the comparison of sequence similarity in high-dimensional space. This paper measures the similarity of two short texts from the perspective of space curve, and solves the problem that the short text cannot be effectively combined with the word order structure. Finally, according to the principle that the smaller the time warping distance is, the higher the similarity degree will be, the time warp distance will be converted into the similarity value between [0, 1]. Through the experimental test, compared with other algorithm models, the algorithm model in this paper has a better effect on the calculation of similarity of short texts with lexical ambiguity.

Introduction to Relevant Technologies

1.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is an open-source pre-training language model put forward by Google in Oct. 2018 as well as the first actual model of deep bidirectional language representations for unsupervised learning. A wide range of text corpuses were used for training to get text representations containing rich semantic information. And then, fine tuning was conducted in the specific downstream NLP task. Finally, these representations were applied in downstream tasks. This model adopted the training modes of "randomized mask" and "next sentence prediction" in pre-training.

The randomized mask training task is as shown in Figure 1. The single character in the randomized mask text sequence forces the model to predict covered characters through consistent learning. For example, if the text sequence "a cat sits on the bed" is processed into "a cat sits on the [MASK]", the covered character is replaced with [MASK] in 80% cases and a random character in 10% cases and remains unchanged in 10% cases. The advantage of such processing lies in that the model can be forced to predict the characteristics of the covered character through learning of understanding characteristics of characters in the context.

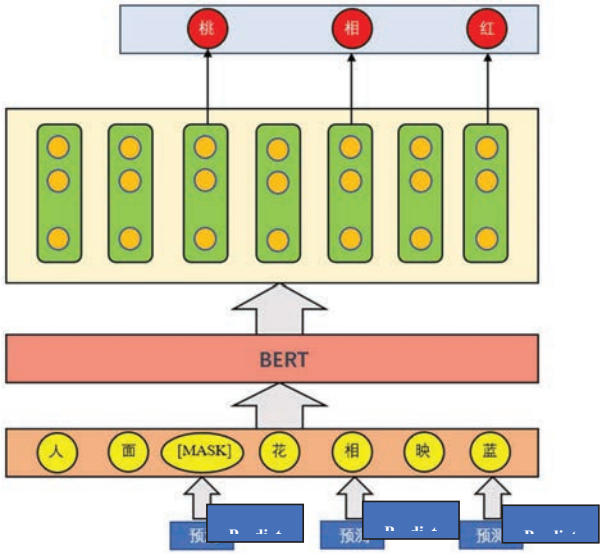


Figure 1 BERT mask pre-training task.

In the next sentence pre-training task, a two-category model is pre-trained to learn the relationship between sentences. The specific training method is as follows: Each item of the sample data contains two sentences and one label value. When the label value is 0, it means that there is not a contextual relationship between two sentences. When the label value is 1, it means that there is a contextual relationship between two sentences.

The input of BERT model is the combination of Token Embeddings, Segment Embeddings and Position Embeddings of two text sentences. Token Embeddings are presented with WordPiece embeddings and Token vocabulary. The first Token is a special category embedded into CLS. SEP Token is used for segmenting two sentences. Segment Embeddings are used for distinguishing two sentences and distinguishing which characters belong to sentence A and which characters belong to sentence B. Position Embeddings are the position vector which is used for marking the position of each character in the sentence. Input of the model is as shown in Figure 2:

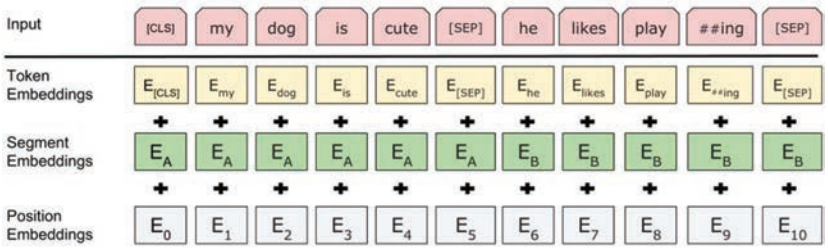


Figure 2 BERT input representation.

The subject of the BERT model is the Encoder of the Transformer. BERT can be considered as the pileup of multi-layer Transformer Encoders.

Output of BERT model is a multi-dimensional eigenvector. Characteristic extraction is one of the functions of BERT model, that is, one paragraph of text can be input into a well-trained model for processing to output a multi-dimensional eigenvector which can be used for presenting the semantic information of the whole text. Generally, this sentence vector is from the last layer or the second last layer of the model or the accumulated output result of the last layer and the second last.

The most central part of BERT model is the Encoder layer of the Transformer. And, the most central part of the Transformer Encoder is the Attention mechanism. The Attention mechanism mainly aims to enhance representation of semantic characteristics of the target character by learning information of characters in the context of the target character.

The Attention mechanism mainly involves three concepts: Query, Key and Value. The target character is deemed as Query (Q). Other characters in the context of the target character are deemed as Keys (K). The character vector corresponding to each character is deemed as Value (V). The weight is calculated through matching of Query and Key. And, the output is gotten based on the weight value, that is the enhanced semantic representation of the target character. The specific formula is as shown in (1):

$$Attention(Q, K, V) = \mathrm{softmax}(\mathrm{sim}(Q, K))V \tag{1}$$

When processing the model, each character in the text should be processed with enhanced semantic vector representation. Each character is deemed as Query. The enhanced semantic vector of the character vector is gotten via weighting of semantic information of other character vectors. This processing mechanism is also called Self-Attention.

Multi-head Self-Attention refers to the process where the enhanced semantic vector of each character in different semantic spaces is processed with linear combination to form an enhanced semantic vector which has the same length as the vector of the original character. The working principle of the Multi-head Self-Attention mechanism is as shown in Figure 3:

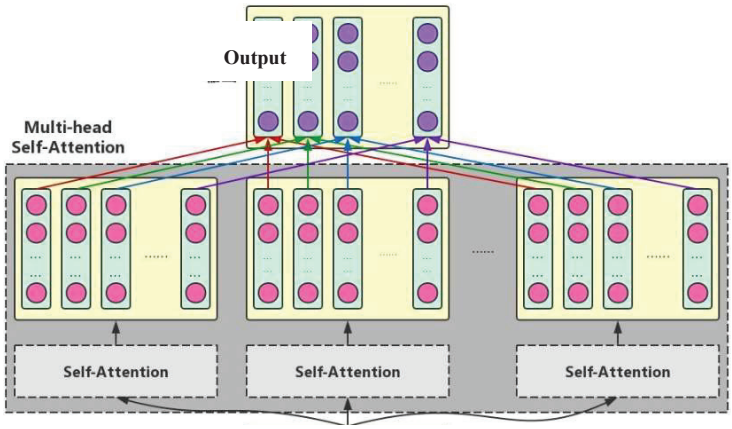


Figure 3 Working principle of Multi-head Self-Attention mechanism.

By adding three key operations to Multi-head Self-Attention, the Encoder structure of the Transformer is formed:

(1) Residual Connection: The output of the last layer and the output of the current layer are stacked, and the stacking result is used as the output of

the next layer. In such a way, loss of characteristic information of text characters during upward processing can be reduced.

(2) Layer Normalization: The output of one layer in the network is processed into a value between [0,1].

(3) Linear conversion: The enhanced semantic vector of each character is processed with two times of linear conversion by guaranteeing that the vector dimension before and after conversion is consistent, so that the characteristic representation capacity of the model can be enhanced.

The structure of the Transformer Encoder is as shown in Figure 4:

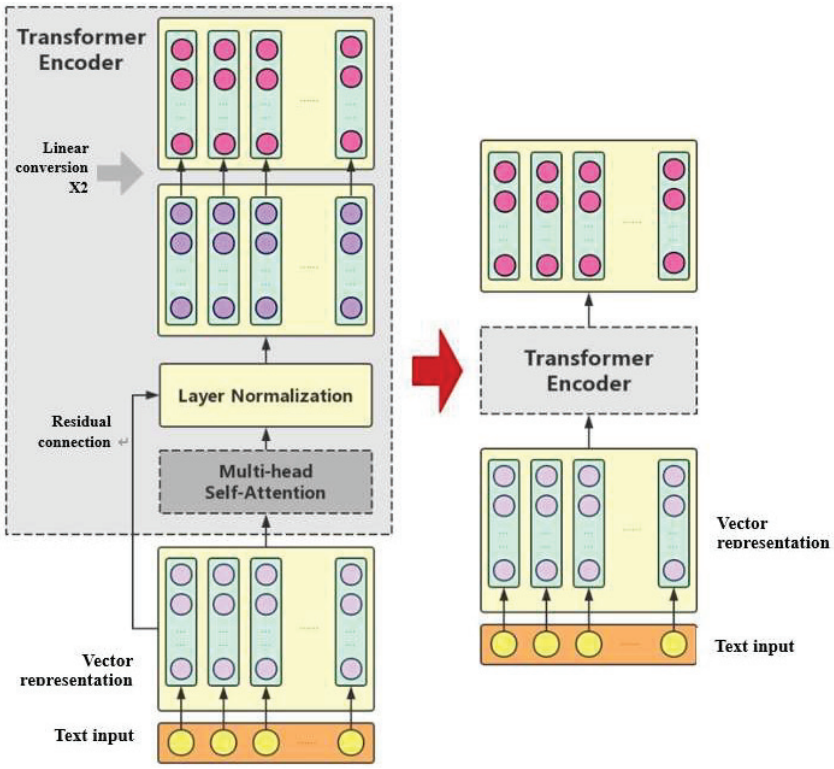


Figure 4 Structure of transformer encoder.

BERT model has many advantages over other language models, especially when processing short text sequences. When learning the deep semantic information of characters in the text sequence, BERT model is forced by randomized mask operation to predict the covered character in combination with the context of the text sequence and the semantic information in the

context of the covered character. After verification with 11 NLP tasks, it is found that this model can process lexical ambiguity well. However, RNN model and LSTM model which are used for processing sequence information involve unidirectional training and cannot realize multi-layer stacking, so training is too slow.

The Attention mechanism in BERT can solve the unidirectional information flow problem well and can extract the relationship between one character and other characters in the whole sentence, regardless of the order. Transformer can solve the problem that multiple layers cannot be stacked in training, that is, the contextual semantic information can be understood in combination with the sentence, parallel execution can be realized during pre-training of the model and multiple layers can be stacked, accelerating the model training speed.

CTW Algorithm

Canonical Time Warping (CTW) [7–9] algorithm, also known as time warping algorithm. It is the generalization of Canonical Correlation Analysis (CCA) in motion spatio-temporal comparison, mainly used for sequence alignment operations. CTW expands the previous research work on CCA in two ways: combining CCA with dynamic time warping (Dynamic Time Warping, DTW) algorithm; expanding CCA by allowing local spatial deformation. In popular understanding, CTW can be regarded as DTW in a multi-dimensional space, and the distance calculated by the CTW algorithm is called the time warping distance, which can be used to measure the similarity of spatial curves.

(1) CCA

Canonical Correlation Analysis (CCA) is a commonly used algorithm in data mining and a classic multivariate statistical analysis method that can extract common features from a pair of multivariate data. The essence of CCA is to select several representative comprehensive indicators (linear combinations of variables) from the two sets of random variables, and use the correlation between these indicators to express the correlation between the original two sets of variables. The specific measurement index is the typical correlation coefficient, and the calculation formula is shown in (2):

$$\rho_{i,j} = \frac{Cov(X_i, X_j)}{\sqrt{Var(X_j)}\sqrt{Var(X_i)}}, Cov(X_i, X_j) - E(X_i)E(X_j) \qquad (2)$$

Amongst them, X_i and X_j represent the i-th and j-th elements of n-dimensional random variables $(X_1, X_2, \ldots, X_n)$, $cov(X_i, X_j)$ represents the covariance of X_i and X_j, $E(X_i)$ represents the expectation of X_i, $Var(X_i)$ represents the variance of X_i. Assuming two sets of variables $X \in R^{p\times q_x}$, $Y \in R^{n\times q_y}$, CCA will find a linear combination, such that:

$$J_{cca}(V_x, V_y) = \|V_x^T X - V_y^T Y\|_F^2 \; s.t. \; V_x^T X X^T V_x = V_x^T X X^T V_x = I_p \tag{3}$$

Amongst them, $V_x \in R^{p\times q_x}$ is the projection matrix of variable X (similar to V_y), I_p is the identity matrix, and the typical variable pair is not correlated with $(V_x^T X, V_y^T Y)$ and other lower-order typical variables. Each continuous canonical variable pair achieves the maximum correlation with the previous pair of orthogonal canonical variables. Formula (3) has a closed solution in dealing with generalized eigenvalue problems.

(2) CTW

As mentioned above, CTW is a combination of CCA and DTW. CTW solves the three main shortcomings of the existing methods on the basis of DTW: CTW provides a feature weighting layer to adapt to different modes (such as capturing data in the forms of video and motion); CTW expands DTW by combining monotonic functions to allow more flexible time warping; unlike DTW, which usually incurs secondary costs, CTW has linear complexity. In order to obtain a CTW cost function, the cost function of the DTW algorithm can be rewritten as the following formula (4):

$$J_{dtw}(W_x, W_y) = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} w_i^{xT} w_j^y \|x_i - y_j\|^2 = \|XW_x^T - YW_y^T\|_F^2 \tag{4}$$

Amongst them, $W_x \in \{0,1\}^{m\times n_x}$ and $W_y \in \{0,1\}^{m\times n_y}$ are the binary selection matrix of alignment sequence X and sequence Y, and W_x, W_y are coded align the path.

In order to adapt to the changes of different types and objects, CTW adds a feature selection mechanism (CCA) on the basis of DTW to reduce the dimensionality of the sequence, so that this transformation allows alignment operations on time series of different dimensions. Specifically, a linear transformation (V_x^T, V_y^T) is added on the basis of the minimum square cost function of DTW. CTW effectively combines DTW and CCA by minimizing

the result of formula (5):

$$J_{\mathrm{ctw}}(W_x, W_y, V_x, V_y) = \|V_x^T X W_x^T - V_y^T Y W_y^T\|_F^2 \tag{5}$$

Amongst them, $V_x \in R^{p\times q_x}$, $V_y \in R^{p\times q_y}$, $p \leq min(q_x, q_y)$, the space warping is parameterized by projecting the sequence to the same low-dimensional coordinate system. V_x and V_y will distort the sequence in time according to the constraints to achieve the best time alignment. In order to remain CTW unchanged for shift, rotation and scaling, the following constraints need to be added: $XW_x^T I_m = 0_{q_x}$, $YW_y^T I_m = 0_{q_y}$; $V_x^T X D_x X^T V_x = V_y^T Y D_y Y^T V_y = I_p$; $V_x^T X W Y^T V_y$ is a diagonal matrix.

CTW expands the operation of CCA by adding time alignment and expands the operation of DTW by allowing feature selection and dimensionality reduction mechanisms for aligning sequences of different dimensions. The efficient combination of CCA and DTW is realized by minimizing the formula (5), which can be used to align high-dimensional sequences in time and space.

BERT + Time Warping Distance Similarity Algorithm Model

First, the BERT model was pre-trained with a wide range of Chinese text corpuses, and then trained BERT model was used to extract characteristics of short texts input to output one multi-dimensional text eigenvector. The time warping distance between text eigenvectors was calculated with the time warping distance calculation method. At last, the time warping distance was converted into similarity between short texts by analyzing the designed weight function. The model realization procedure is as shown in Figure 5:

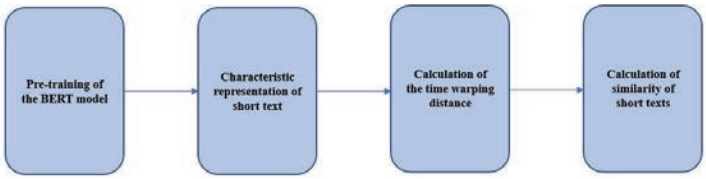


Figure 5 Flow chart of BERT + time warping distance similarity algorithm model.

BERT Model Pre-Training

In normal cases, before processing short texts with the BERT model, the model needs to be pre-trained with a wide range of text corpuses. And, if short news texts are processed, training with news corpuses will achieve

a better processing effect. BERT model needs to be trained for about 4 days in the distributed environment of 4∼16 TPUs. The existing experiment hardware resources were not sufficient, so, this paper directly adopted the Chinese corpus model (chinese_L-12_H-768_A-12) generated through Chinese Wikipedia corpus training provided in an open-source way by Google on Nov. 3, 2018. This model structure contains 12 Transformer Encoder layers, 768 hidden neurons, 12 Mutli-Heads and 110M network parameters.

Table 1 File description of BERT model (chinese_L-12_H-68_A-12)

No.	File Name	File Description
1	bert_model.ckpt.data-00000-of-00001	Model weight
2	bert_config.json	Model parameters
3	bert_model.ckpt.index	Model information
4	bert_model.ckpt.meta	Model meta information
5	vocab.txt	Vocabulary

Short Text Characteristic Representation

In terms of short text characteristic representation, the specific processing procedure is as shown in Figure 6:

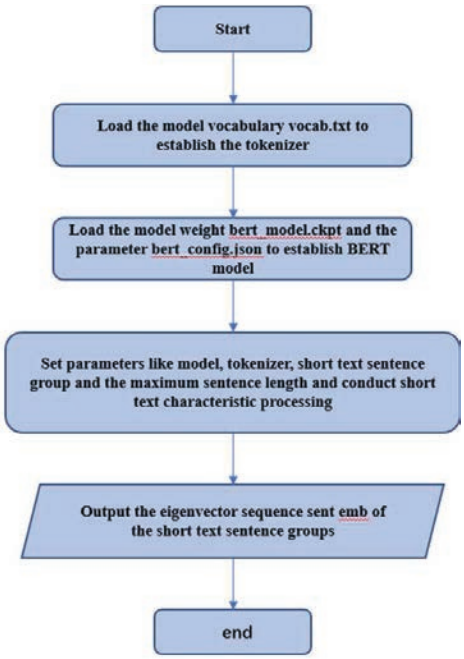


Figure 6 BERT short text characteristic process flow.

Since 768 hidden neurons are used in the BERT model, each piece of short text output after processing is corresponding to 768-dimension eigenvectors. During processing, generally short text arrays are input, and 768-dimension eigenvector sequence is output. The eigenvector corresponding to each text can be deemed as the comprehensive characteristic representation of semantic information of characters in the text and the syntactic structure of the text.

Calculation of the Time Warping Distance

The algorithm model put forward in this paper simulates short text eigenvector by forming curves connecting the sequences of points in the space and compares the similarity of curves to reflect the similarity of short texts. Since the observation capacity of DTW algorithm in the high-dimensional space is restricted, the performance effect is poor. So, CCA and DTW were combined to align with CTW algorithm featured with excellent operation performance in the high-dimension space sequence to calculate the warping distance of text eigenvectors. The time warping distance calculation process is as follows:

Calculation process of time warping distance

Input: Text eigenvector X, text eigenvector Y

Output: Minimum cost (time warping distance) J_{ctw}

Start

Initialize $V_x = I_{d_x}, V_y = I_{d_y}$

Looping execution: Calculate W_x and W_y with the dynamically planned thinking; To make two sequences align with each other, vector $V^T = [V_x^T, V_y^T]$ was extracted from typical variable $V_x^T X$ and $V_y^T Y$ as the generalized eigenvector of the designating data b, and the formula (6) should be met:

$$\begin{bmatrix} 0 & XWY^T \\ YW^TX^T & 0 \end{bmatrix} V = \begin{bmatrix} XD_xX^T & 0 \\ 0 & YD_yY^T \end{bmatrix} V\wedge \tag{6}$$

Until: The maximum value of J_{ctw} was gotten

The process was ended

Wherein, $D_x = W_x^T W_x$, $D_y = W_y^T W_y$, $W = W_x^T W_y$. It is a non-convex optimization issue about the alignment matrix (W_x, W_y) and projection matrix (V_x, V_y) that the minimum value of J_{ctw} is calculated with the calculation formula (formula (5)) of J_{ctw}. During calculation, DTW algorithm was used to alternatively solve W_x and W_y, and CCA was used to optimize the calculation space projection b and V_y.

Calculation of Similarity of Short Texts

After calculating the time warping distance of two short text eigenvectors, the result should be converted into the similarity value between [0, 1], and it observes the law that as the time warping distance is smaller, the similarity value is bigger. Through experiment and analysis of the time warping distance, a weight function which can be used to convert the time warping distance into the similarity was designed. It is as shown in formula (7):

$$Sim(s_1, s_2) = \frac{e^{3*(2-ctw(s_1,s_2))}}{1 + e^{3*(2-ctw(s_1,s_2))}} \tag{7}$$

Wherein, $ctw(s_1, s_2)$ means the time warping distance between the eigenvectors of short text s_1 and s_2. $Sim(s_1, s_2)$ is the final similarity result. In the experiment, the max_len (maximum text length) was extracted from the eigenvector function of BERT model for dynamic setting to adjust implications of different lengths of short texts on the experiment result.

Experiment and Result Analysis

To verify the effectiveness of the similarity algorithm model in the context of lexical ambiguity and normal contexts, some literatures [8] and short texts released on Tencent News and Baidu News in recent years were used for similarity calculation, and the contrast experiment was carried out by comparing it with the method of BERT + cosine similarity in the literature [11].

Experimental Environment

Environment of the similarity experiment: The similarity calculation and the contrast experiment were finished with the Windows operating system. Python was used as the main programming language. Hardware resources include 16G memory, GTX 1650 graphics card and Intel i7-9750 processor. Python3.7 was applied. To use BERT to process short texts in Python, bert4keras library package needs to be installed.

Experimental Data

Since this experiment attaches importance to comparison of the similarity of short text sentence pairs, there are special requirements for the experimental data, and the two-category text similarity dataset cannot be directly used. So,

the similarity of two short texts is distinguished with the value between 0–1. 200 groups of short texts (each group contains three short texts whose length is less than 200) were selected from literatures [8] and Tencent News and Baidu News released in recent years as the experimental data. Some experimental data are shown in Table 2:

Table 2 Short text data of BERT + time warping distance similarity experiment

Group	Short Text Data
Group 1	s_1: I like drinking pitaya juice
	s_2: The fruit I like is apple
	s_3: I like using Apple mobile phone
Group 2	s_1: Alipay Ant Credit Pay supports advance consumption
	s_2: Ant Credit Pay offers an advance consumption quota monthly
	s_3: We can spend the money we make through working hard to buy whatever we want
Group 3	s_1: Fuji apples produced by Shandong Liangmin Fruit Products are superb
	s_2: Shaanxi Fuji apples have won so much praise, because they are two sweet
	s_3: Mountain Fuji has always been a holy mountain in eyes of world travelers
	s_1: Students who agree can sit down
Group 4	s_2: Please raise your hand if you agree
	s_3: I like comfortable life
	s_1: He pushed his little brother over
Group 5	s_2: His little brother pushed him over
	s_3: Brother fell into water accidently
	s_1: He put the umbrella on the desk
Group 6	s_2: He holds a beautiful umbrella
	s_3: He wears a long scarf

Wherein, short text data of group 1, group 2 and group 3 contain ambiguous characters (marked with the underline), and group 4, group 5 and group 6 do not contain ambiguous characters.

Experiment Steps

Similarity between s1 and s2 and similarity between s1 and s3 was mainly calculated in this experiment with BERT + time warping distance method put forward in this paper by following the process in Figure 7, and the calculated similarity result was compared with the result generated with method of BERT + cosine similarity in the literature [11].

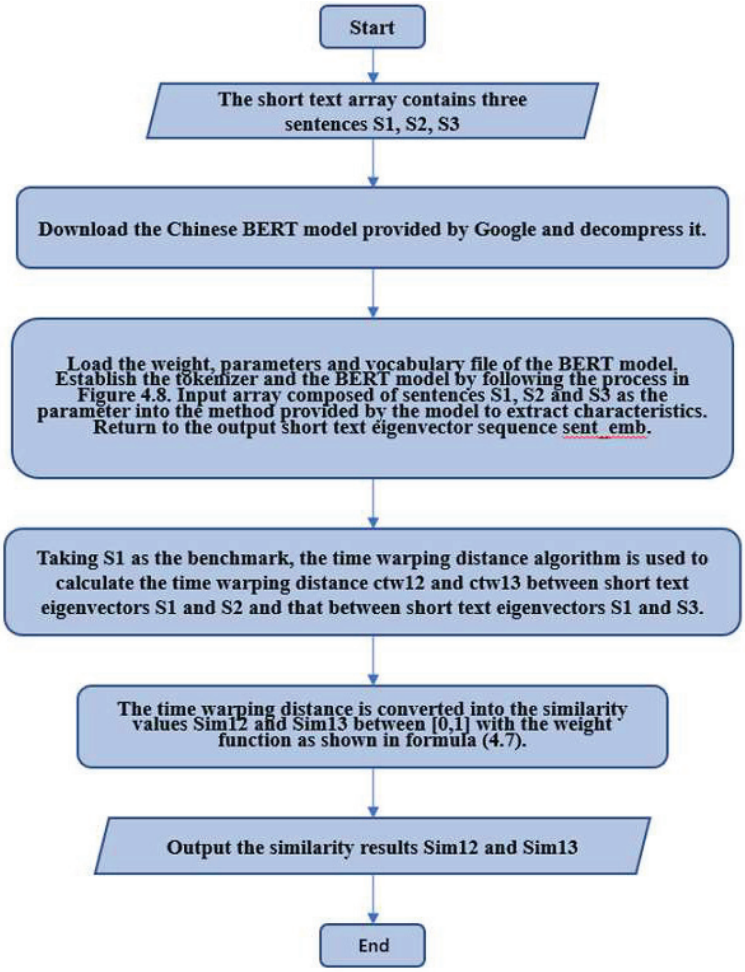


Figure 7 Flow process of BERT + time warping distance similarity algorithm.

Taking the first group of short text data in Table 2 as an example, three short texts were input into the characteristic extraction function in the form as shown in Figure 8 by following the process flow in Figure 7:

```
sentence=["I like drinking pitaya juice", "The fruit I like is apple", "I like using Apple mobile phone"]
```

Figure 8 Format of data input.

One eigenvector sequence whose length is 3 was output after processing with BERT model. The eigenvector curve corresponding to three short texts is as shown in Figures 9–11:

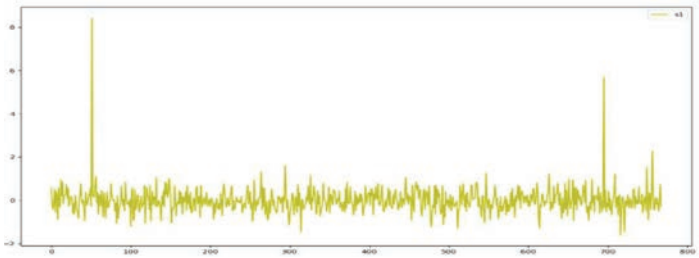


Figure 9 Eigenvector curve of text 'I like drinking pitaya juice'.

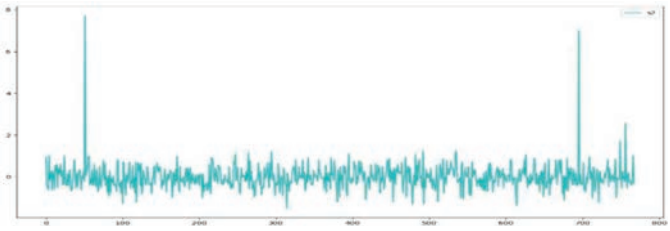


Figure 10 Eigenvector curve of text 'The fruit I like is apple'.

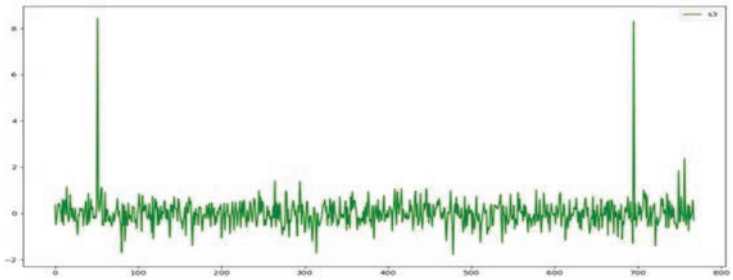


Figure 11 Eigenvector curve of text 'I like using Apple mobile phone'.

After processing with BERT model, each short text was converted into 768-dimension eigenvector, and then the time warping distance between the eigenvector of S1 and that of S2 and the time warping distance between the eigenvector of S1 and that of S3 were calculated by following the

time warping distance calculation process flow as described in Section 2.3. Calculation results are as follows: $\mathrm{ctw}_{12} = 1.5994, \mathrm{ctw}_{13} = 2.0329$.

And then, the time warping distance was converted into similarity with the formula (4.7). The result was $Sim_{12} = 0.7688, Sim_{13} = 0.4753$.

Experimental Result

The time warping distances between short text sentence pairs calculated with the BERT + time warping distance similarity algorithm are as shown in Figure 12:

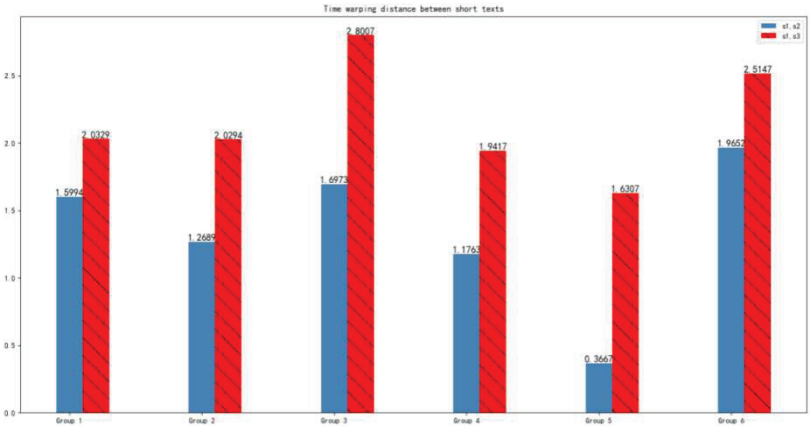


Figure 12 Histogram of time warping distance between short texts

Six groups of short text data calculated were from Table 2. To facilitate comparison of experimental results, relatively similar short texts found through artificial analysis were put at the first piece place and the second piece place of each group. It is obviously found from the above figure that the time warping distance between s_1 and s_2 is smaller than that between s_1 and s_3 of each group. As a whole, the result follows the principle that as the similarity is higher, the time warping distance is smaller. And, the similarity of texts of different experimental data arrays is different too. Figure 10 shows that the difference of time warping distance between s_1 and s_2 and that between s_1 and s_3 in different groups of data are different, conforming to the actual situation that the similarity between short texts is different.

The similarity of 200 groups of experimental data was calculated, and the calculation result was compared with the similarity result calculated with the method put forward in literature [11]. To facilitate representation, the

similarity calculation method put forward in this chapter is abbreviated as BERT-CTW. The method put forward in literature [11] is abbreviated as BERT-Cosine. Some experimental results are as shown in Table 3:

Table 3 BERT + time warping distance similarity algorithm experiment result comparison table

Group	Short Text Data	Comparative Sentence	BERT-Cosine	BERT-CTW
Group 1	s_1: I like drinking pitaya juice	s_1, s_2	0.8782	0.7688
	s_2: The fruit I like is apple	s_1, s_3	0.7842	0.4753
	s_3: I like using Apple mobile phone			
Group 2	s_1: Alipay Ant Credit Pay supports advance consumption	s_1, s_2	0.8868	0.8996
	s_2: Ant Credit Pay offers an advance consumption quota monthly	s_1, s_3	0.6972	0.4779
	s_3: We can spend the money we make through working hard to buy whatever we want			
Group 3	s_1: Fuji apples produced by Shandong Liangmin Fruit Products are superb	s_1, s_2	0.8349	0.7126
	s_2: Shaanxi Fuji apples have won so much praise, because they are two sweet	s_1, s_3	0.7507	0.0830
	s_3: Mountain Fuji has always been a holy mountain in eyes of world travelers			
Group 4	s_1: Students who agree can sit down	s_1, s_2	0.7685	0.9220
	s_2: Please raise your hand if you agree	s_1, s_3	0.6561	0.5436
	s_3: I like comfortable life			
Group 5	s_1: He pushed his little brother over	s_1, s_2	0.9691	0.9926
	s_2: His little brother pushed him over	s_1, s_3	0.8009	0.7517
	s_3: Brother fell into water accidently	s_1, s_2	0.8513	0.5260
Group 6	s_1: He put the umbrella on the desk	s_1, s_3	0.7460	0.1759
	s_2: He holds a beautiful umbrella			
	s_3: He wears a long scarf			

Experimental Analysis

Through analyzing similarity results as shown in Table 3, it is found that if we compare the similarity result between s_1 and s_2 and that between s_1 and s_3 of each group of short text without taking into consideration the difference scope, the BERT-Cosine method and the BERT-CTW method put forward in

this paper can get $\mathrm{Sim}(s_1, s_2) > \mathrm{Sim}(s_1, s_3)$, which means that both methods can distinguish the similarity between s_1 and s_2 and that between s_1 and s_3 of short texts well in both the case with ambiguous vocabulary (group 1–group 3) and normal cases (group 4–group 6).

In view of the specific difference of similarity of each group of short texts, the mean value of differences of similarity between s_1 and s_2 and that between s_1 and s_3 with the BERT-Cosine method is about 0.12, which means that the similarity of short texts calculated with this method is relatively high, dissimilar short texts cannot be distinguished well through calculation with this method, whilst similar texts and dissimilar texts can be distinguished well with the BERT-CTW method put forward in this paper. For example, in group 1, the semantic information of s_1 means "I like drinking fruit juice", that of s_2 means "I like each a fruit (apple)". Both of them convey the information of "liking fruits", so their similarity is high. However, s_1 emphasizes that the person "likes juice", while s_2 emphasizes that the person "likes fruits". So, there are subtle distinctions based on high similarity. The calculation result with BERT-Cosine method is 0.8782, and that with BERT-CTW method is 0.7680. There is the word "apple" in s_3, but the whole semantic information is that the person "likes using the mobile phone of one brand named Apple". So, the information of s_3 is quite different from s_1 and s_2. The calculation result with BERT-Cosine method is 0.7842 which is smaller, showing an insignificant difference. But the calculation result with BERT-CTW method is 0.4748, showing a bigger difference and a better distinguishing effect on dissimilar short texts. And, in the similarity comparison between ambiguous words "Ant Credit Pay" and "Fuji" in three short texts, the similarity result of s_1 and s_3 of dissimilar short texts calculated with BERT-CTW method is 0.4780 and 0.0846 respectively, so the BERT-CTW method can distinguish other three groups of dissimilar short texts in normal cases well.

Conclusions

To make clear implications of lexical ambiguity on similarity between short texts, this paper puts forward one BERT + time warping distance short text semantic similarity algorithm model. Short text eigenvectors were output after processing with the BERT model; the time warping distance between short text eigenvectors was calculated with CTW algorithm; At last, the time warping distance was converted into similarity between short texts by analyzing the designed weight function. Experimental analysis reveals that this algorithm model can explore characteristic information of ambiguous

vocabulary represented in the current short text, effectively calculate the similarity of short texts and can distinguish more accurately short texts containing ambiguous vocabulary than BERT-Cosine method. However, limited to experimental resources, news corpuses were not used in this paper to pre-train BERT model. If Chinese news corpuses are used to pre-train the BERT model, the text eigenvector representation effect may be better, and the calculated similarity result will conform to the actual situation better. The mask training of BERT model only aims to single Chinese character. If the semantic unit in the text is segmented in advance, exploration for characteristics of ambiguous vocabulary may be further improved when masking single semantic unit at random during masking training.

References

[1] GUO Shenguo, Xing dandan. Sentence similarity calculation based on word vector and its application research[J]. Modern Electronics Technique, 2016, 39(13):99–102.

[2] Liu Xinting, CAI Xiaodong. Sentence similarity calculation based on word vector and Chinese frame net[J]. Journal of Guilin University of Electronic Technology, 2017, v. 37; No. 153(06):67–70.

[3] Jeffrey Pennington, Richard Socher, Christopher Manning. GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, October 25–29, 2014, Doha, Qatar. c 2014 Association for Computational Linguistics.

[4] Matthew E. Peters, Mark Neumann, Mohit Iyyer, et al. Proceedings of NAACL-HLT 2018, pages 2227–2237 New Orleans, Louisiana, June 1–6, 2018. c 2018 Association for Computational Linguistics.

[5] Nguyen, Hien & Duong, Phuc & Cambria, Erik. (2019). Learning short-text semantic similarity with word embeddings and external knowledge sources. Knowledge-Based Systems. 182. 10.1016/j.knosys.2019.07.013.

[6] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[7] Zhou F, Torre F D L. Canonical Time Warping for Alignment of Human Behavior[C]// Advances in Neural Information Processing Systems 22: Conference on Neural Information Processing Systems A Meeting Held December. Curran Associates Inc. 2009.

[8] Feng Zhou, Fernando De la Torre. Generalized Canonical Time Warping[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 38(2):1–1.

[9] George Trigeorgis, Mihalis A. Nicolaou, Stefanos Zafeiriou, et al. Deep Canonical Time Warping[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE, 2016.

[10] Li X, Li Q. Calculation of Sentence Semantic Similarity Based on Syntactic Structure[J]. Mathematical Problems in Engineering, 2015, 2015:1–8.

[11] Yin Qingshan, Li Rui, Yu Zhilou. A method and process of similarity matching of short text based on deep learning Bert algorithm: CHINA, 18797489.2[P].2019-09-29.

[12] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2019). TinyBERT: Distilling BERT for Natural Language Understanding. ArXiv, abs/1909.10351.

[13] Gao Huang, Chuan Guo, Matt J. Kusner, Yu Sun, et al. Supervised Word Mover's Distance[C]. Neural Information Processing Systems Conference, 2019.

[14] Rakthanmanon T, Campana B, Mueen A, et al. Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping[C]// Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM, 2012.

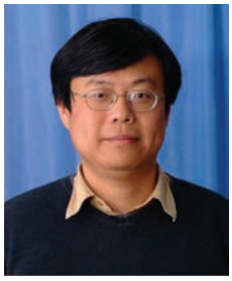
[15] Wei J, Ren X, Li X, et al. NEZHA: Neural Contextualized Representation for Chinese Language Understanding[J], 2019.

[16] Trigeorgis G, Nicolaou M A, Zafeiriou S, et al. Deep Canonical Time Warping[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.

[17] Nguyen, Hien & Duong, Phuc & Cambria, Erik. (2019). Learning short-text semantic similarity with word embeddings and external knowledge sources. Knowledge-Based Systems. 182. 10.1016/j.knosys.2019.07.013.

Biographies

Shijie Qiu is a Master student at Hubei University of Technology since Autumn 2019. He received his B.Sc. in Computer Engineering in summer 2018. Mazen is currently completing a Master's Degree in Computer Science at the School of Computer Science, Hubei University of Technology. His work centers on computer applications and artificial intelligence algorithm.



Yan Niu is a professor at the school of computer science, Hubei University of technology. He received his B.Sc. in Wuhan University of Technology. Then he went to study at the Southern Institute of technology in New Zealand. Mainly part-time, including national 863 plan project evaluation experts, executive director of microcomputer Professional Committee of Hubei computer society, etc., and obtained a number of international certification certificates.

Jun Li Associate Professor, Department of Software Engineering, School of Computer Science, Hubei University of Technology, with research interests in natural language processing, information security, and network protocol analysis. Presided over 1 provincial-level scientific research project, and 1 comprehensive reform project of industry-university cooperation between the Ministry of Education and Microsoft Corporation.

Xing Li received his M.Sc. degrees in computer Engineering from Hubei University of Technology, Starting from 2020, he has been working at China Communications Services Sci and Tech Co., Ltd. His research directions are Natural language processing and neural network.