A Chinese Short Text Semantic Similarity Computation Model Based on Stop Words and TongyiciCilin

Tang Shancheng, Bai Yunyue, Ma Fuyu
Communication and Information Institute
Xi'an University of Science and Technology
Xi'an, China
tangshancheng@21cn.com

***Abstract*—Short text similarity computing plays an important role in natural language processing, and it can be applied to many tasks. In recent years, there are lots of researches getting important results on natural language processing. Although there are some good results in English, there is no major breakthrough in Chinese. Different from the proposed methods, we reserve the Stop words in the training dataset of word vector for Chinese characteristics, and add the TongyiciCilin to the training data of the short text semantic similarity computation model. We compared the effect of Word2vec and Glove methods in our model. We use the Chinese short text semantic similarity dataset which is designed by Chinese grammar experts. The results show that the accuracy of the model is improved by 2%-3% by retaining Stop words in word vector training data and adding TongyiciCilin to training data. The accuracy of our model is better than Baidu short text similarity calculation platform on the same testing dataset.**

Keywords—Semantic Similarity; Chinese; Stop Words; TongyiciCilin

I. INTRODUCTION

In recent years, with the deep learning image processing [1, 2], speech recognition [3] and other fields made major breakthroughs. Natural Language Processing has become an important research area for the future. The text semantic similarity computing has been an important research topic in the field of natural language processing. It can be applied to many tasks such as machine translation [4], paraphrasing problem [5], automatic question and answer [6], text classification [7], Information retrieval [8], and so on. Due to the complexity and abstraction of the text, the semantic similarity computing is still confronted with great challenges.

There are many short text semantic similarity computing models based on the deep learning method. We divide these models into two types: single-granularity short text semantic similarity computing model and multi-granularity short text semantic similarity computing model. The single-granularity short text semantic similarity computing models are expressed by words or sentences as vectors, and the text similarity value is obtained by computing the similarity of the vectors, such as DSSM [9, 10], CLSM [11], and LSTM-RNN [12] model. The multi-granularity text semantic similarity computing models are based on the single-grayscale text semantic similarity computing models, and the representation of the text is not only words or sentences, but also the combination of these features, such as the MultiGranCNN model [13], the uRAE model [14], and the MV-LSTM model [15].

Some of the models mentioned above have achieved good results in dealing with English texts, but there is still no breakthrough in Chinese processing [16-17]. For English texts, removing the Stop words can really improve the computational results of semantic text similarity, but it is counterproductive in Chinese texts processing. Chinese texts have many unique features compared with other languages, such as: 1) Relative to English and other Indo-European languages, Chinese does not have strict grammar [18-20]. 2) We need to get the word segmentation of texts before processing Chinese texts, [21-23]. Different word segmentations of the same Chinese texts lead to different meanings. 3) To understand Chinese, we need to analyze context [18-20, 24-26], and Stop words can provide contextual information.

In view of the above features of Chinese, we propose to retain the Stop words in the preprocessing text job. The TongyiciCilin is added to the training dataset to further improve the accuracy of the model. The experiment result shows that the accuracy of our model is better than Baidu short text similarity calculation platform on the same testing dataset.

II. STOP WORDS AND SYNONYMS FORESTS

Converting the short texts into word sequences is the primary work in Natural Language Processing. The previous researchers believe that function words only play the role in the structure of texts and do not represent actual significance, such as quantifiers, pronouns, position words and interjections [26]. In addition, there are some words that appear frequently in the entire dataset, and have roughly equal probabilities in each document, such as "的,了,我" and so on. We put these words together as Stop words [27]. Most researchers believe that these words have no effect on semantic text similarity

computing and should be removed. For Chinese texts, the computational complexity is reduced, but the accuracy of the model is reduced for Chinese semantic similarity calculations. The Stop words provide some semantics in Chinese sentences, for example:

A: "我吃饭了"

B: "他吃饭"

If we remove the Stop words in A and B, the two sentences will become "吃饭". Therefore, the Stop words in Chinese cannot be arbitrarily removed.

TongyiciCilin is compiled by Mei Jiaju et al. And it is a dictionary that includes a synonym for the words and a considerable number of similar words [28]. It divides the words into lines according to the meaning of the words, each of which has the same semantic and strong correlation, such as: No. Bh06A42 = "豆荚" "豆角儿" "豆角", there are three synonyms in the same line; No. Bh06A32 = "番茄" "西红柿", there are two synonyms in the same line [28]. In TongyiciCilin, a group data is composed of some words in the same line, and the semantic similarity between these words is marked as 10. These word pairs are added to the training dataset so that the model can identify synonymous substitutions of sentences. Such as "我喜欢吃西红柿" and "我喜欢吃番茄", the similarity of these two sentences should be 10.

III. MODEL

A. Text Preprocessing and Pre-training

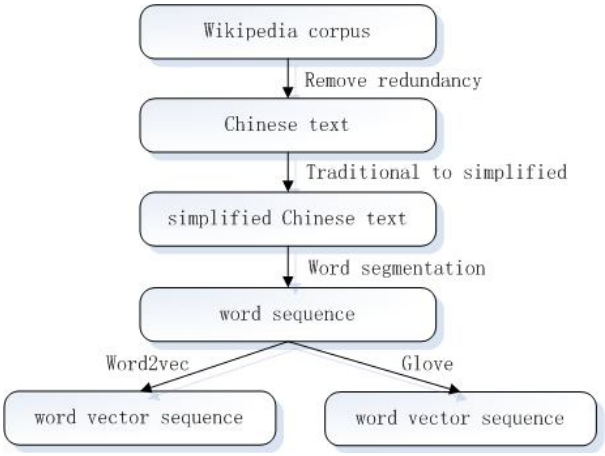

Fig. 1. Flow chart of preprocessing

We uses the Wikipedia Chinese data set to do pre-training. Because the Wikipedia corpus contains URLs, traditional Chinese and various identifiers, we use the Wikiextractor tool to handle the corpus and use Opencc to convert texts into simplified Chinese texts. The word segmentation technology has been quite mature, we use the Jieba tool to divide the texts into many word sequences. Usually, the proposed models will remove Stop words after this step, but we think that Stop words should be reserved, for Chinese is different from other languages, and the Stop words play an important role in Chinese sentences.

After the text preprocessing, the next step is pre-training on the dataset. We train the word vectors by two methods: Word2vec and Glove respectively. Word vectors are divided into four categories: The first category is Word2vec word vector that removes Stop words. The second category reserves Stop words for the Word2vec word vector. The third category is the Glove word vector which removes Stop words. The fourth category reserves Stop words for the Glove word vector.

B. Model Architecture

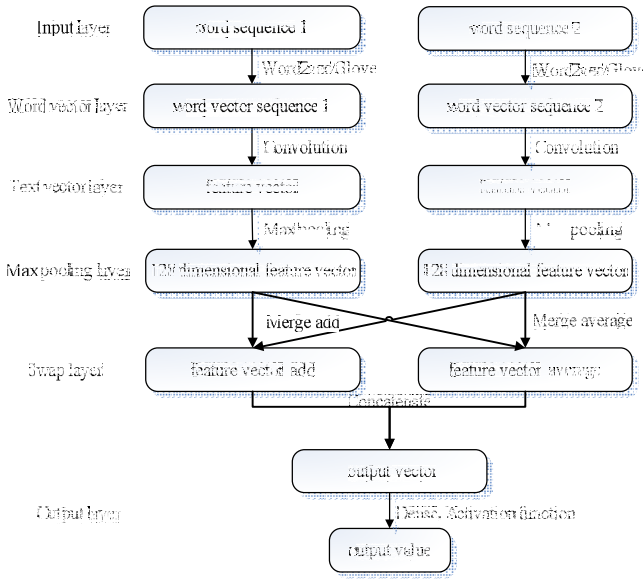

Fig. 2. Structure diagram of the model

Our model is composed of two parallel convolutional neural networks, and the overall architecture is shown in Figure 2. The input layer inputs the text dataset which needs to be trained, and the short Chinese texts are processed into word sequences. 2) The word vector layer transforms the word sequences into word vectors which can be trained by the model. This section maps the word sequences to the pre-trained word vector sequences. 3) The convolutional layer obtains the context information from the words in the convolution kernel, thus obtaining the semantic information of the words and sentences. 4) The max-pooling layer selects the best features that represent the text information. Those are also called features at the sentence level. 5) The swap layer adds and averages the feature information of the two texts to avoid errors due to the processing of two sentences by two parallel models. 6) The output layer concatenates the two text vectors to generate the values of the text semantics similarity finally. Our model calculates similarity in this part by four dense layers. The activation function in the first three dense layers is Relu, and the activation function in the last layer uses the nonlinear softmax function.

C. Activation Function and Loss Function

In the neural network the activation functions can be added to some nonlinear factors. They make neural networks have

the ability to solve complex problems. After convolution operations, we usually use the tanh function as the activation function, which is defined as equation (1):

$$\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \quad (1)$$

Its value range is (0, 1), and the convergence rate is faster than the sigmoid function. The output is centered at 0.

After the model merges two text vectors, four dense layers are designed. Since the Relu function converges quickly, the first three layers use it as activation function, defined as equation (2):

$$y = \begin{cases} x, & x < 0 \\ 0, & x \geq 0 \end{cases} \quad (2)$$

Compared to tanh, the Relu function is linear and non-saturated, so it can converge quickly in SGD. Compared to the softmax and the tanh function that need to calculate index, Relu can get the activation value with only one threshold.

In the output layer, softmax function is adopted as activation function. The softmax function is defined as equation (3):

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad (3)$$

The essence of the softmax function is to compress (map) any real vector of an K dimension into another K dimension of vector, where the value of each element is between (0, 1) in the vector. Then we can deal with multi-classification tasks. Our model is equivalent to doing multi-classification tasks, so we use the categorical crossentropy function as the loss function corresponding to the activation function softmax.

IV. Data Sets

A. Data sets of training word vectors

The data sets for Chinese short text semantic similarity computing are relatively small. There are Wikipedia and Sogou data sets which are free and large. Our model uses the public Chinese data set of the Wikipedia of 1.75G. We can get the Chinese data of 1G after removing redundant data, and can get about 800M Chinese data removed Stop words.

B. Data sets of training model

We divide the training data into two types: one is the training data without TongyiciCilin, and the other is the training data including TongyiciCilin. According to the Chinese grammar rules [29-30], the training data contains paraphrases, negative sentences or unrelated sentences. For a pair of sentences with identical semantics, the similarity value is marked as 10. For a pair of sentences with irrelevant semantics, the similarity value is marked as 5. For a pair of sentences with opposite semantics, the similarity value is marked as 0. At last, 2747 pairs of sentences were arranged. Therefore, in order to construct the semantic unrelated sentence pairs, we extract some sentences from Wikipedia data, and these sentences have same keywords as the main sentences. These sentences form sentence pairs with the main sentences. The irrelevant and related sentence pairs were screened manually. At last, 12769 pairs of valuable sentences were selected.

The training data of the TongyiciCilin is based on the word pairs in TongyiciCilin. Words with same meaning are selected from TongyiciCilin. Two words form a word pair. Finally, the original 12769 pairs of sentences and 58,620 pairs of words are trained by the model.

C. Data sets of testing model

We selected 1703 pairs of sentences to test our model.

V. Conclusion Experiment and Analysis

A. Parameter setting

The dimension of the word vectors in the model is 200. The smaller dimension of the word vectors cannot fully express the semantic information of words, and the larger dimension will bring difficulty in calculation which causes the training speed to be too slow. The general Dropout ranges from 20%~50%. We select dropout parameter as 0.2. If the Dropout parameter is smaller, it cannot suppress the over-fitting. If the parameter is larger, it will lead to that the model is less learning. We set 200 as the maximum length of sequences. The model uses 80% of the total data as the training data set, and the remaining 20% as the validation data set. The final iteration number of the model is chosen 50 times. In the experiments, we found that the accuracy is improved and the loss is declining in the earlier iterations. However, after this period, the iterations do not improve the results of training, but sometimes it will decline. The activation function and the loss function in the model are described in detail in Section 2.3.

B. Training results and analysis

With the model described above, the results of our training are shown in Table 1. From Table 1 we get the following conclusions: 1) the results of training and testing with Stop words are generally better than those of removing Stop words. 2) TongyiciCilin gives a slight boost to the results. Specifically, Without Stop words, the TongyiciCilin is added to the training set and the training results are improved. With Stop words, the TongyiciCilin is added to the training set and the testing results have improved. 3) The result of word vectors trained by Glove is better than Word2Vec.

From the above analysis, we can conclude that the training results with TongyiciCilin and Stop words are the best. And the testing results with TongyiciCilin and Stop words are the best also.

The accuracy of training is shown in Figure 3, and the loss of training is shown in Figure 4. The red lines in the pictures are fitted by the tool, and the actual values are showed as the curves that are distributed dimly around the red lines. As shown in Figure 3, the accuracy of the first 20 iterations is improved very quickly, but the effect is not obvious after the 20 iteration. And the accuracy fluctuates up and down. At the time of the forty-fifth iteration, the accuracy decreased

significantly. As shown in Figure 4, the loss drops rapidly during the first 20 times, but the effect is not obvious and gradually approaches 0 after 20 iterations. Therefore, our model of iterations 20 times is reasonable.

In this section, we also test our testing dataset with the API of Baidu short text similarity calculation platform (http://ai.baidu.com/tech/nlp/simnet?castk=LTE%3D). There are 1703 pairs in our test dataset, and Baidu's testing result has 83 errors. Our model goes beyond the Baidu model.

TABLE I. SUMMARY OF RESULTS

Errors /Total	Non SW Non TC	Non SW TC	SW Non TC	SW TC [a]
Training Word2Vec	179/10216	61/46891	4/10216	16/46891
Testing Word2Vec	66/1703	74/1073	49/1073	41/1073
Training Glove	48/10216	30/46891	1/10216	40/46891
Testing Glove	34/1073	67/1073	45/1073	41/1073

[a] SW= Stop words, TC=TongyiciCilin

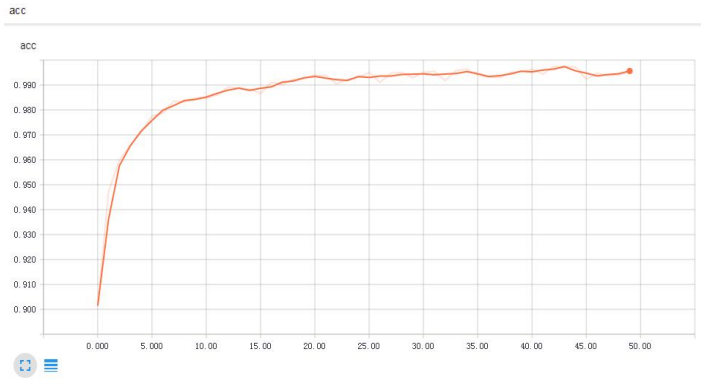

Fig. 3. Accuracy curve of training

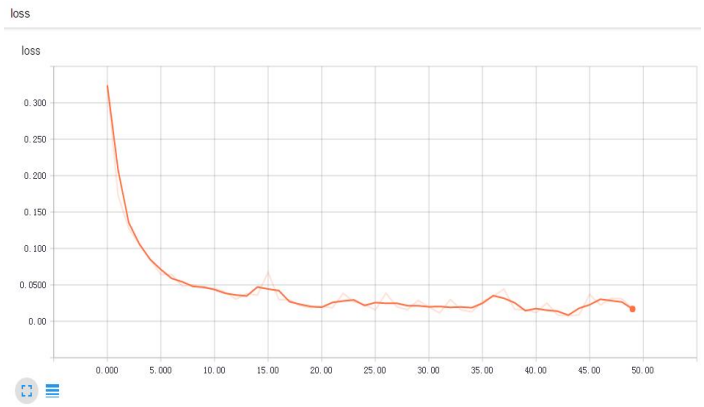

Fig. 4. Loss curve of training

VI. CONCLUSIONS

In this paper, we evaluate and improve the previous training methods according to the grammatical features of Chinese [31-35], and further improve the accuracy of the short Chinese semantic text similarity computing model. The main contributions are summarized in the following three aspects: Firstly, unlike previous models, our model retains the Stop words in the training dataset. Chinese differs from other languages, the Stop words in Chinese texts have semantic information. Secondly, the words with the same meaning in TongyiciCilin are added to the training dataset of the model, so the model can identify the words with similar meaning, and this can improve the accuracy of semantic similarity calculation. Thirdly, the effect of Glove and Word2Vec on pre-training word vector is compared, and Glove gets better result. However, the addition of Stop words and TongyiciCilin has greatly improved the accuracy of training and testing in our model, which is beyond the previous model. This computing model of short Chinese semantic text similarity can be applied to many tasks, such as machine translation, paraphrasing problem, automatic question and answer, text classification, information retrieval and so on. New approach will bring new ideas for the Natural Language Processing field.

ACKNOWLEDGMENT

This article is sponsored by "Scientific Research Program Funded by Shaanxi Provincial Education Commission (Program NO. 2013JK1079)".

REFERENCES

[1] LeCun Y, BengioH Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1995, 3361(10): 1995.

[2] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, USA, 2015: 1-9.

[3] Abdel-Hamid O, Mohamed A, Jiang H, et al. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. 2012 IEEE international conference on Acoustics, speech and signal processing. Kyoto, Japan, 2012: 4277-4280.

[4] Brown P F, Pietra V J D, Pietra S a D, et al. The mathematics of statistical machine translation: Parameter estimation. Computational linguistics, 1993, 19(2): 263-311.

[5] Dolan W B, Brockett C. Automatically constructing a corpus of sentential paraphrases. Proceedings of the Third International Workshop on Paraphrasing, Jeju Island, Korea, 2005: 9–16.

[6] Xue X, Jeon J, Croft W B. Retrieval models for question and answer archives. Proceedings of the Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. Singapore, 2008: 475-482.

[7] Zeng D, Liu K, Lai S, et al. Relation Classification via Convolutional Deep Neural Network. Proceedings of the COLING. Dublin, Ireland, 2014: 2335-2344.

[8] Li H, Xu J. Semantic matching in search. Foundations and Trends in Information Retrieval, 2014, 7(5): 343-469.

[9] Huang P-S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data. Proceedings of the 22nd ACM international conference on Conference on information and knowledge management. Amazon, India, 2013: 2333-2338.

[10] Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, Li Deng. Modeling Interestingness with Deep Neural Networks .Microsoft Research.

[11] Shen Y, He X, Gao J, et al. A latent semantic model with convolutional-pooling structure for information retrieval. Proceedings of the 23rd ACM international conference on

Conference on information and knowledge management. New York, USA, 2014: 101-110.

[12] Palangi H, Deng L, Shen Y, et al. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(4): 694-707.

[13] Yin W, Schütze T, Hinrich. MultiGranCNN: An Architecture for General Matching of Text Chunks on Multiple Levels of Granularity. Proceedings of the 53rd Annual meeting of the association for computational linguistics, Beijing, China, 2015: 63-73.

[14] Socher R, Huang E H, Pennin J, et al. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. Proceedings of the Advances in Neural Information Processing Systems, Granada, Spain, 2011: 801-809.

[15] Wan S, Lan Y, Guo J, et al. A deep architecture for semantic matching with multiple positional sentence representations. Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 2835-2841.

[16] Li Yan. Research on Analysis and Computation Methods for Short Text with Deep Learning. Doctoral Dissertation of Beijing University of Science and Technology. 2016.

[17] Xiaoyang Chen. Deep Learning for Short Text Semantic Similarity Measures. Master Thesis of Beijing Institute of Technology. 2015

[18] Wu Zuoyan, Wang Yu. A New Measure of Semantic Similarity Based on Hierarchical Network of Concepts. Journal of Chinese Information Processing. 2014, 28(2): 37-44. G

[19] Dong Zhengdong, Dong Qiang. HowNet and Chinese Studies. Contemporary linguistics. 2001, 3(1): 33-44.

[20] Dong Zhengdong, Dong Qiang, Hao Changling. Theoretical Findings of HowNet. Journal of Chinese Information Processing. 2007, 21(4): 3-10.

[21] Zhu Yanhui, Liu Jing, Xu Yeqiang, et al. Chinese word segmentation research based on Conditional Random Field. Computer Engineering and Applications, 2016, 52(15):97-100.

[22] Feng Guohe, Zhen Wei. Review of Chinese Automatic Word Segmentation. Library and Information Service. 2011, 55(2):41-45.

[23] Huang Changning, Zhao Hai. Chinese Word Segmentation: A Decade Review. Journal of Chinese Information Processing. 2007, 21(3): 8-19.

[24] Jiang Zhaozhong. Chinese Words Segmentation Based on Context and Stop words. Master Dissertation. 2010.

[25] Hang Maoyuan, Lu Zhengding, Zou Chunyan. Chinese Word Segmentation Based on Language Situation. Mini- Micro Systems. 2005, 26(1): 129-133.

[26] Cui Caixia. Research on the Effect of Stop Words Selection on Text Categorization. Journal of Taiyuan Normal University (Natural Science Edition). 2008, 7(4): 91-93.

[27] Tian Jiule, Zhao Wei. Words Similarity Algorithm Based on TongyiciCilin in Semantic

[28] Web Adaptive Learning System. Journal of Jilin University (Information Science Edition). 2010, 28(6): 602-608.

[29] Shi Yuzhi. Chinese Grammar. Beijing: The Commercial Press. 2015.

[30] Huang Borong, Li Wei. Contemporary Chinese language. Peking University press. 2016.

[31] LI Wenqing, Sun Xin, Zhang Changyou, Feng Ye. A Semantic Similarity Measure between Ontological Concepts. Acta Automatica Sinica. 2012, 38(2): 229-235.

[32] Jia Wenjuan, He Feng. Research of Chinese Ontology Learning Based on HowNet. Computer Technology and Development. 2011, 21(6): 77-81.

[33] Wu Zuoyan, Wang Yu. A New Measure of Semantic Similarity Based on Hierarchical Network of Concepts. Journal of Chinese Information Processing. 2014, 28(2): 37-44.

[34] Xu Ge, Wang Houfeng. The Development of Topic Models in Natural Language Processing. Chinese Journal of Computers. 2011, 34(8): 1423-1236.

[35] Liao Zhifang, Zhou Guoen, Li Junfeng, Liu Fei, Cai Fei. A Chinese Short Text Similarity Algorithm Based on Semantic and Syntax. Journal of Hunan University (Natural Sciences). 2016, 43(2): 135-140.