http://dx.doi.org/10.14257/ijdta.2016.9.10.32

Chinese Sentence Similarity Computational Model Based on Multi-Features Combination

Peiying Zhang, Qiuming Li and Huayu Li

College of Computer & Communication Engineering, China University of Petroleum (East China), Qingdao shandong 266580, China
E-mail: 25640521@qq.com

Abstract

Combined with the issue of single direction of the solution of the existing sentence similarity algorithms, a Chinese sentence similarity computational model based on multi-features combination was proposed. The approach combines word overlap similarity, word order similarity, dependency relationship similarity, semantic similarity, structure similarity, sentence similarity, and keyword distance similarity to calculate the similarity between sentences, using the weight to describe the contribution of each feature of the sentence, and then gets a better experiment result. Experimental results shows that this approach can fully describe the features of the sentence, therefore can improve the sentence similarity computation accuracy.

***Keywords**: sentence similarity; word overlap similarity; word order similarity; dependency relationship similarity; semantic similarity*

1. Introduction

The model of Chinese sentence similarity is of great significance in the natural language processing fields. At present sentence similarity computational algorithm plays an increasingly important role in text-related research and applications in areas such text mining (Li *et al.*, 2006); example-based machine translation to find the best similar sentence in the instances of sentences; frequently asked questions to match the sentence among the question sentences. Previous works are confined to literal meaning of the request sentence that users input. With the rapid development of the semantic calculation technology and natural language processing technology, researchers began to pay more attention to the real intention behind the sentence that users want to convey, that is, seeing the essence through the phenomenon and returning the most satisfactory search results to the users. Sentence similarity computation has become an important research hot pot in the field of Chinese information processing and has a wide range of applications in information retrieval, text categorization, question answering system and machine translation, and so forth.

The calculation of sentence similarity generally can be divided into three aspects: syntactic similarity, semantic similarity, and pragmatic similarity. Due to the rapid development of natural language processing technology, syntactic parsing has been applied to the sentence similarity computation [5, 7], pragmatic similarity is the highest goal and it is quite difficult to realize at present. Semantic similarity computation has been advanced which mainly utilizes the taxonomy knowledge base such as “HowNet”, “Cilin” and “WordNet”. “HowNet” is a notable common sense knowledge base with concepts represented by Chinese and English words for describing the objects and revealing the relationships between concepts and the relationships between their attributes as the basic contents [8]. “WordNet” is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a

ISSN: 2005-4270 IJDTA

distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations.

2. Related Works

Recently literature on sentence similarity has shown abundantly proposed methods [1-2, 5-7, 9-10, 14-15, 17]. Li *et al.* [1] took advantage of information from a structured lexical database and from corpus statistics to calculate the semantic similarity between two sentences, the use of a lexical database enables the method to model human common sense knowledge and the incorporation of corpus statistics allows the method to be adaptable to different domains. P. Achananuparp *et al.* [2] proposed an approach which takes into account the variability of natural language expression. In literature [2], the author evaluate fourteen existing text similarity measures which have been used to calculated similarity score between sentences in many text applications. Xiao Li *et al.* [5] analyzes the sentence constituent, and then through analysis convert sentence similarity into words similarity on the basis of syntactic structure, then convert words similarity into concept similarity through words disambiguation, finally realize the semantic similarity comparison. Xiong Jing *et al.* [6] proposed an approach that introduces a dependency syntactic tree similarity computation method based on multi-features. The method studies many feature of dependency syntactic tree including word and word's part of speech of each node and the dependency type between them, then utilizing all the feature to measure the sentence similarity. Ferreira, R *et al.* [7] presented a three-layer sentence representation and a new measure to compute the degree of similarity between two sentences. The three layers are: (i) the lexical layer, which encompasses lexical analysis, stop words removal and stemming; (ii) the syntactic layer, which performs syntactic analysis; and (iii) the semantic layer that mainly describes the annotations that plays a semantic role. Hien T. Nguyen *et al.* [9] introduced a novel method for measuring semantic similarity between two sentences. The originality of the method is that it explores named entities and their co-reference relations as important indicators for measuring the similarity. This method exploits WordNet, Wikipedia, and Brown Corpus and learns a classifier using features. Li Bin *et al.* [10] put forward a method that based on semantic dependency relationship analysis to compute sentence similarity, and its experimental results is satisfied. WANG Rong-bo *et al.* [14] proposed a similarity measure method of Chinese sentence structures for example-based Chinese to English machine translation. ZHOU Fa-guo *et al.* [15] presented an improved sentence similarity method which includes the extraction of keywords and the induction of synonyms in sentence similarity definition. Based on this, a question answer system based on FAQ is implemented. C. P. Cheng *et al.* [17] proposed an improved method of similarity comparison. The semantic tree of sememe is constructed according to the description of entity conception in the HowNet, the semantic similarity of sememe is computed based on the relative positions in the sememe tree. The experimental results show that the proposed method is much closer to the people's comprehension to the meanings of the sentences.

3. Sentence Similarity Computational Model

3.1. Word Overlap Similarity Measure

Word overlap measure is a combinatorial similarity measure that computes similarity score based on the number of words shared by two sentences. Simple word overlap is defined as the number of words that appear in both sentences normalized by the sentence's length. IDF overlap uses inverse document frequency to normalize the result [2]. Phrasal overlap measure can be defined as the relation between phrases length and their document frequencies. Traditional word overlap measure treats a sentence as a bag of words and doesn't take into account the difference between single words and multiword phrases. Because n-word overlap is much rarer than single word overlap, and its effectiveness is more important than single word overlap's effectiveness, thus m phrasal n-word overlaps are defined as a non-linear function as follows:

$$overlap_{phrase}(S_1,S_2) = \sum_{i=1}^{n}\sum_{m} i^2 \tag{1}$$

where m is the number of *i-word* phrases which appear in both sentences S_1 and S_2. The equation can be normalized by the sum of sentence length and applying the hyperbolic tangent function to minimize the effectiveness of outliers [2]. The word overlap similarity can be defined as the formula (2):

$$Sim_1(S_1, S_2) = \tanh \frac{overlap_{phrase}(S_1, S_2)}{|S_1| + |S_2|} \tag{2}$$

where S1 and S2 denote the two sentences, |S1| and |S2| denote the length of two sentences, tanh() is the function of hyperbolic tangent.

3.2. Word Order Similarity Measure

The word order similarity measure can be defined as the degree of dissimilarity between sentence S1 and S2. Let's take a particular case to illustrate the importance of word order. For two sentences:

S1: A yellow cat is running after a black dog.

S2: A yellow dog is running after a black cat.

These two sentences contain exactly the same words and most words appear in the same order. The only difference is that cat appears before dog in sentence S1, and the dog appears before cat in sentence S2. Since the two sentences contain the same words, any methods based on "bag of words" can't distinguish the two sentences. However, it is obviously for a human interpreter that S1 and S2 are only similar to some extent. The difference between S1 and S2 is the result of word order permutation. Therefore any effective computational model for sentence similarity must take into account the impact of the word order.

Sentences containing the same words but in different orders may result in very different meanings. Human can easily understand the meanings of sentences which own different words orders, but it is difficult to distinguish the meanings through computer programs. However, incorporate the words order information into the computational methods for understanding the natural language sentences is a difficult challenge. This is the reason why most previous studies do not tackle the word order similarity between sentences. In this section we use the method which proposed by literature [9] to measure the word order similarity between sentences.

For a pair of sentences, we denote the joint word set by T. Recall the above two example sentences, their joint word set is:

T = {A yellow cat is running after a black dog}

For each word in S_1 and S_2, a unique index number has been assigned respectively. Then we form word order vectors to be r_1 and r_2 respectively, for these sentences based on T. To show how to assign entry values for a word order vector, we take S_l as example:

Case 1: if a word w in T appears in S_l, we fill its entry value in r_l with the corresponding index in S_l.

Case 2: if w does not appears in S_l, we compute word-sim between w and in turn words S_l and keep the word w' having the highest word-sim with w; if the highest word-sim is greater than a preset threshold, the entry value of w in r_l is set to the index of w'; otherwise, it is set to zero.

Note: The word-sim is the word similarity between two words, and it can be calculated by the algorithm [8].

Taking the above two sentences as example, we can compute the word order vectors of S1 and S2 as follows:

$$r_1 = \{1,2,3,4,5,6,7,8,9\}$$

$$r_2 = \{1,2,9,4,5,6,7,8,3\}$$

The word order similarity between two sentences can be finally attributed to the similarity computation between two vectors r_1 and r_2, we use the formula to measure their similarity as follows.

$$Sim_2(S_1, S_2) = 1 - \frac{\|r_1 - r_2\|}{\|r_1 + r_2\|} \tag{3}$$

Form the above formula, we can make the conclusion that word order similarity is determined by the normalized difference of word order.

3.3. Dependency Relationship Similarity Measure [6]

The dependency relationships in the sentence can be derived by the Stanford Parser, which is a statistical parser that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb. The two sentences can be represented by two relationship collections, the similarity of the two relationship collections is based on the similarity of two relationships. We use P (H/H-POS, C/C-POS, D) to describe one relationship. What it indicates is that word C is dependent on word H, H-POS denotes the part-of-speech of the word H, C-POS denotes the part-of-speech of the word C, and D denotes the dependency type between C and H. The five elements in P have different importance and should be assigned to different weight ratios.

According to the theory of dependency grammar, one word can only depend on a specific word but it can have many words depended on it. Therefore in P, the word C is more important than the word H. In terms of other views, one word has a few kinds of POS but on POS includes lots of words, hence the word itself is obviously more important than its POS. Finally, the dependency type D depends not only on the words but also on the words' POS, so the importance of D is between the words and their POS. Therefore, we get the order of weight for the five features: C>H>D>C-POS>H-POS.

Suppose there are two relationships of dependency, represented by P_1= {H_1/H_1-POS, C_1/C_1-POS, D_1} and P_2= {H_2/H_2-POS, C_2/C_2-POS, D_2}, due to the fact that the weight coefficient of these five features are different, we use binary representation method to measure the similarity. We sort these five features by their weights, and we can get a binary number $(bbbbb)_2$, which is ranged from 0 to 31. 0 means the two relationships are totally different and 1 means the two are exactly the same. Based on this binary representation method, we can define the similarity between P_1 and P_2 as:

$$SR(P_1, P_2) = \frac{(bbbbb)_2}{(11111)_2} \tag{4}$$

Suppose there are two set of relationships A= (a_1, a_2, …, a_n) and B= (b_1, b_2, …, b_m). Based on the similarity between P1 and P2, we can define the similarity between sentence relationship collections as follows:

$$Sim_3(S_1, S_2) = \frac{\sum_{i=1}^{n} \max_{1 \le j \le m} SR(a_i, b_i) + \sum_{i=1}^{m} \max_{1 \le i \le n} SR(b_i, a_i)}{n + m} \tag{5}$$

3.4. Semantic Similarity Measure

The semantic similarity between two sentences can be defined as the similar degree of their meanings. Li *et al.* [1] suggest a semantic-vector approach to compute sentence similarity. Sentences are transformed into feature vectors with distinct words from both examined sentences as a feature set T. The semantic similarity between two sentences is computed as a cosine similarity between feature vectors of the two sentences.

$$Sim_{Li}(S_1, S_2) = \frac{S_1 \times S_2}{\|S_1\| \times \|S_2\|} \tag{6}$$

Semantic similarity of sentences can be finally attributed to the word semantic similarity. This paper calculate word semantic similarity with the help of "HowNet" platform [8] which is a common sense knowledge base with concepts represented by Chinese and English words for describing the objects and revealing the relationships between concepts and the relationships between their attributes. The similarity between word W and sentence S can be defined as the maximum value between the word W and all words in S, the formula is as follows:

$$WSSim(W, S) = \max_{W_i \in S} Sim(W, W_i) \tag{7}$$

where Sim(W, Wi) denotes the similarity between word W and word Wi. The semantic similarity between two sentences can be defined as follows.

$$Sim_4(S_1, S_2) = \frac{\sum_{W_i \in S_1} WSSim(W_i, S_2) + \sum_{W_i \in S_2} WSSim(W_i, S_1)}{|S_1| + |S_2|} \tag{8}$$

where |S| denotes the number of words which sentence S contains.

3.5. Structure Similarity Measure

The structure similarity between two sentences mainly reflects the similar extent in their structures. This paper use the approach which proposed by [14] to measure the structure similarity, the main ideas of this approach is that: For two part-of-speech (POS) sequences of two sentences, combined with their weight value of their POS to match, with the purpose of getting the optimal matching results. The structure similarity between two sentences can be defined as follows:

$$Sim_5(S_1, S_2) = 2 \times \sum_{i=1}^{C} \frac{W_i}{1 + \sum_{j=1}^{D} W_j} \Big/ \sum_{k=1}^{E} W_k \tag{9}$$

where C denotes the number of common nodes in two part-of-speech sequences, Di denotes the number of the non-merge nodes in a loop, E denotes the total number of nodes in two part-of-speech sequences. Wi denotes the weight value of i-th shared nodes, Wj denotes the weight value of j-th nodes in a loop, Wk denotes the weight value of k-th nodes in all of the POS nodes.

3.6. Sentence Length Similarity Measure [15]

The sentence length similarity can reflect the similar degree of two sentences in their lengths. The sentence length similarity can be defined as follows:

$$Sim_6(S_1, S_2) = 1 - abs(\frac{Len(S_1) - Len(S_2)}{Len(S_1) + Len(S_2)}) \tag{10}$$

where abs() denotes the absolute value, Len(Si) denotes the number of words which sentence Si contains.

3.7. Keyword Distance Similarity Measure

The keyword distance similarity can be measured by the distance between two shared keywords, the formula can be defined as follows:

$$Sim_7(S_1, S_2) = 1 - abs(\frac{SameDis(S_1) - SameDis(S_2)}{Dis(S_1) + Dis(S_2)}) \tag{11}$$

where abs() denotes the absolute value, SameDis(Si) denotes the distance of common keywords in Si, if the keyword appears more than one time, take the maximal distance as its value. Dis(Si) denotes the distance between the left and the right keywords which is non-repeating in Si. If the keyword appears more than one time, take the minimal distance as its value. More detail can refer to literature [15].

3.8. Sentence Similarity Measure Based on Multi-Features

The sentence similarity refers to the matching extent in semantic of two sentences which is a real number between the value of [0, 1]; the greater the value, the greater the similarity of the two sentences. If the two sentences are the same in semantics, its value is 1; if the two sentences in semantics are totally different, its value is 0. Denote two sentence by S1 and S2, the sentence similarity can be defined as follows.

$$Similarity(S_1, S_2) = \sum_{i=1}^{7} \lambda_i \times Sim_i(S_1, S_2) \tag{12}$$

where λ_i is an adjustable parameter, and satisfied the equation $\sum_{i=1}^{7} \lambda_i = 1$.

4. Experiments and Analysis

4.1. MSRP Data Set

This paper uses the notable publicly-available sentence pair sets to evaluate the performance of the sentence similarity measures. The data sets of Microsoft Research Paraphrase Corpus (MSRP) consists of 5801 pairs of sentence automatically constructed from various news sources on the web. These sentences were separated into training data (4076 sentences pairs) and test data (1725 sentence pairs). We translated some of sentence pairs into Chinese and tested their similarity to test our proposed approach of sentence similarity.

4.2. Experimental Results

We chose some of sentence pairs from the MSRP, and translated them into Chinese sentences. Compared with the approach of TF-IDF and approach of semantic dependency to measure our proposed algorithm. The experimental results are shown in Table 1.

Table 1. Experimental Results

The Algorithm	Number of Sentence Pairs	Correct Number	Accuracy (%)
TF-IDF	100	43	43
Semantic Dependency [10]	100	82	82
Our Approach	100	88	88

Experimental results show that our approach is better than other methods, due to the reason that it take word overlap similarity, word order similarity, dependency relationship similarity, semantic similarity, structure similarity, sentence length similarity, and keyword distance similarity into considerations. Through the analysis of wrong sentence pairs we found that the reason which result in wrong computation is these sentence pairs contain unknown words, own complicated structures, contain lot of words, and contain negative words and so on. This is the main reason to influence the sentence similarity computation accuracy.

5. Conclusion

Based on the analysis of the existing sentence similarity algorithm, the sentence similarity algorithm based on multi-features fusion was proposed, take advantage of the word order, dependency relationship, semantics, structure, sentence length, and keyword distance to measure the sentence similarity. The originality of our method is that it explores multi-features and their contributions as important indicators for measuring the similarity. The experimental results show that all of those have contributions to the sentence similarity. In comparison with others, our method is straightforward and quite easy to implement, while giving state-of-the-art results. Our future work will focus on investigating their contribution by assigning different weight value, the weight value can be automatic learned utilize the convolution neural network and other deep learning technologies.

Acknowledgments

This work is supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2014FQ018) and "the Fundamental Research Funds for the Central Universities" of China University of Petroleum (East China) (Grant No.13CX02031A). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] Y. Li, D. Mclean, Z. A. Bandar, J. D. O'Shea and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics", IEEE Trans. Knowledge Data Eng., vol. 28, **(2006)**, pp. 1138-1150.

[2] P. Achananuparp, X. Hu, and X. Shen. "The Evaluation of Sentence Similarity Measures", In Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery. Spring-Verlag, **(2008)**.

[3] R. Levy and C. D. Manning, "Is it harder to parse Chinese, or the Chinese Treebank", ACL 2003, **(2003)**, pp. 439-446.

[4] P. C. Chang, H. Tseng, D. Jurafsky, and C. D. Manning, "Discriminative Reordering with Chinese Grammatical Relations Features", In Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation, **(2009)**.

[5] X. Li and Q. S. Li, "Calculation of Sentence Semantic Similarity Based on Syntactic Structure", Mathematical Problems in Engineering, **(2015)**, pp. 1-8.

[6] X. Jing, L. Y. Tong and Y. Dong, "Dependency syntactic tree supported sentence similarity computing", Information Technology Journal, vol. 12, no. 20, **(2013)**, pp. 5685-5688.

[7] R. Ferreira, R. D. Lins and S. J. Simske, "Accessing sentence similarity through lexical, syntactic and semantic analysis", Computer Speech Language, **(2016)**.

[8] Q. Liu and S. J. Li, "Calculation of lexical semantic similarity based on the "HowNet", in Proceedings of the 3rd Chinese lexical semantics workshop, Taipei, Taiwan, **(2002)**.

[9] H. T. Nguyen, P. H. Duong and T. Q. Le, "A multifaceted approach to sentence similarity", (Eds.): IUKM, LNAI 9376, **(2015)**, pp. 303-314.

[10] L. Bin, L. Ting and Q. Bing, "Chinese sentence similarity computing based on semantic dependency relationship analysis", Application Research of Computers, vol. 12, **(2003)**.

[11] R. Mihalcea, C. Corley and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity", In: AAAI, vol. 6, **(2006)**, pp. 775-780.

[12] J. Oliva, J. I. Serrano, M. D. del Castillo and A. Iglesias, "SyMSS: A syntax-based measure for short-text semantic similarity", Data & Knowledge Engineering, vol. 70, no. 4, **(2011)**, pp. 390-405.

[13] N. X. Bach, N. L. Minh and A. Shimazu, "Exploiting discourse information to identify paraphrases", Expert Systems with Applications, vol. 41, no. 6, **(2014)**, pp. 2832-2841.

[14] W. R. Bo and C. Z. Ru, "A similarity measure method of Chinese sentence structures", Journal of Chinese information processing, vol. 19, no. 1, **(2005)**, pp. 21-29

[15] Z. F. Guo and Y. B. Ru, "New method for sentence similarity computing and its application in question answering system", Computer Engineering and Applications, vol. 44, no. 1, **(2008)**, pp. 165-167

[16] W. He and Y. Wang, "Text representation based on sentence and Chinese text categorization", Journal of the China society for scientific and technical information, vol. 28, no. 6, pp. 839-843

[17] C. P. Cheng and Z.G. Wu, "A method of sentence similarity computing based on HowNet", Computer Engineering & Science, vol. 32, no. 2, **(2012)**, pp. 172-175.

[18] V. Rus, N. Niraula and R. Banjade, "A study of probabilistic and algebraic methods for semantic similarity", In: Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, **(2013)**.

[19] H. S. Nalwa, "Magnetic Nanostructures", American Scientific Publishers, Los Angeles, **(2003)**.

[20] H. V. Jansen, N. R. Tas and J. W. Berenschot, "Encyclopedia of Nanoscience and Nanotechnology", Edited H. S. Nalwa, American Scientific Publishers, Los Angeles, vol. 5, **(2004)**, pp. 163-275.

Authors

Peiying Zhang, obtained his master degree from China University of Petroleum (East China) in 2006. He is currently a lecturer in the college of computer and communication engineering. He is a PhD candidate in Information and Communication Engineering, from the State Key Lab of Networking and Switching Technology in Beijing University of Posts and Telecommunications. His research interests include natural language processing, semantic computing, future internet architecture, network virtualization, and data center network.

Qiuming Li, obtained his bachelor degree from China University of Petroleum (East China) in 2002. He is a master candidate in College of Computer and Communication Engineering, from China University of Petroleum (East China). Her research interests include natural language processing, semantic computing.

Huayu Li, obtained his PhD degree from University of Science and Technology Beijing in 2010. He is currently an associate professor in the college of computer and communication engineering, from China University of Petroleum (East China). His research interests include natural language processing, semantic computing, and database theory.