Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 203475, 8 pages
http://dx.doi.org/10.1155/2015/203475

Research Article

Calculation of Sentence Semantic Similarity Based on Syntactic Structure

Xiao Li[1] **and Qingsheng Li**[1,2]

[1] *School of Computer and Information Engineering, Anyang Normal University, Anyang, Henan 455002, China*
[2] *Institute of Digital Inscriptions on Bones/Tortoise Shells, Anyang, Henan 455002, China*

Correspondence should be addressed to Xiao Li; joylx@163.com

Received 14 November 2014; Revised 8 December 2014; Accepted 1 January 2015

Academic Editor: Chih-Cheng Hung

Combined with the problem of single direction of the solution of the existing sentence similarity algorithms, an algorithm for sentence semantic similarity based on syntactic structure was proposed. Firstly, analyze the sentence constituent, 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, and, finally, realize the semantic similarity comparison. It also gives the comparison rules in more detail for the modifier words in the sentence which also have certain contributions to the sentence. Under the same test condition, the experiments show that the proposed algorithm is more intuitive understanding of people and has higher accuracy.

1. Introduction

Information retrieval has become an effective way for people to access resources, and the effectiveness of retrieval has been an important index that people are most concerned about. Previous retrieval results are confined to literal meaning of the request sentence that users input. With the development of the semantic web technology and natural language processing technology, people began to pay more attention to the real intention behind the sentence that users input, that is, seeing the essence through the phenomenon and returning the most satisfactory search results to the users. The key to this process is the calculation of similarity. Sentence similarity computation has become an important research content in the field of Chinese information processing and has a wide range of applications in information retrieval, text classification, question answering system and machine translation, and so forth.

Sentence similarity computation generally consists of three levels: syntactic similarity, semantic similarity, and pragmatic similarity, in which the pragmatic similarity is the highest goal and it is quite difficult to realize at present. Adding semantic similarity computation on the premise of syntactic similarity can greatly improve the retrieval effect and meet the needs of people.

In this paper, with the help of the function of syntactic analysis and semantic role labeling of LTP platform (Language Technology Platform) of Harbin Institute of Technology [1], similarity calculation method is proposed on the basis of sentence constituent analysis, and the calculation of words similarity based on HowNet is translated into relatively simple calculation of the concept similarity through words disambiguation. The proposed algorithm not only makes up for the defects of lack of semantic of literal matching algorithms, but also simplifies the complexity of the algorithm, and finally improves the precision.

2. Calculation of Words Similarity Based on HowNet

2.1. Calculation of Words Similarity of HowNet. Sentence similarity computation can be finally attributed to words similarity computation. This paper calculates the words similarity with the help of "HowNet" platform [2] which is a common sense knowledge base with concepts represented by Chinese and English words for describing the objects and revealing

```
NO.=037649
W_C=Da
G_C=verb [da3]
S_C=
E_C=˜cao gao, ˜fu gao
W_E=work out
G_E=verb [51work■verb■-0■vt,sobj,ofnpa■21   ]
S_E=
E_E=
DEF={compile|bian ji}
RMK=
```

Box 1: Representation of HowNet concept.

the relationships between concepts and the relationships between their attributes as the basic contents [3].

In “HowNet,” all words are described by one or several “concepts,” and each concept is described by a group of “sememes.” The sememe is used to describe the smallest meaningful unit of a “concept” and each sememe represents a different role. Each concept in HowNet is described with a record as shown in Box 1.

Listed in Box 1 is the word “Da” which means “work out”: “NO.” means concept number; “W_C, G_C, S_C, E_C, W_E, G_E, S_E, E_E” represent the words, part of speech, positive evaluation, examples, English words, English part of speech, positive evaluation in English, English examples, respectively. “DEF” is a semantic representation of sememe collection described by “knowledge description language.” Thus, the description of concepts in HowNet is more complex, which has also brought great difficulties to the calculation of words similarity.

For the two Chinese words W_1 and W_2, if W_1 has m senses (i.e., concepts), $S_{11}, S_{12}, \ldots, S_{1m}$, W_2 has n senses (concepts): $S_{21}, S_{22}, \ldots, S_{2n}$; HowNet provides that the similarity between W_1 and W_2 be the maximum value of various concept similarity; that is,

$$\operatorname{Sim}(W_1, W_2) = \max_{i=1\ldots m, j=1\ldots n} \operatorname{Sim}(S_{1i}, S_{2j}). \qquad (1)$$

The calculation of concept similarity can be finally attributed to the calculation of sememe similarity. The calculation of sememe similarity mainly has three kinds of methods: one is proposed by Zhang and Li [4] which considers the influence of the number of public nodes and depth on similarity; one is the formula of sememe similarity based on the theory of the commonness and individuality proposed by Lin [5]; the other is the method of HowNet calculating the similarity through semantic distance (gotten by a sememe hierarchy tree composed of sememe hyponymy) [3]; that is,

$$\operatorname{Sim}(p_1, p_2) = \frac{\alpha}{d + \alpha}. \qquad (2)$$

In the above formula, both p_1 and p_2 mean sememes, α is an adjustable parameter, and d which is a positive integer is the length of the path of p_1 to p_2 in the sememe hierarchy.

The words included in “HowNet” are divided into two categories: notional words and function words. In actual text, the similarity of notional words and function words is always zero; the similarity computation of notional concept is more complex because it is described in a semantic representation and readers can refer to the article of “HowNet lexical semantic similarity computation” by Liu and Li [3].

Thus, HowNet attributes the similarity problem of two words to the similarity problem of two concepts. But knowing from the relationship between words and concepts, if the relationship between the two is one-to-one, the concept of HowNet can be directly taken as the meaning of the word (i.e., that word is equal to the concept); if the relationship between the two is one-to-many or many-to-many, then the similarity of various concepts of the two words needs to be calculated one by one, and then take the maximum.

2.2. Improvement on Word Similarity Computation of HowNet

2.2.1. Word Disambiguation. Considering the problem of more complex algorithm and large amount of calculation in the second relationship above, this paper has improved this problem.

HowNet only considers two isolated words in the calculation of word similarity. If the word is placed in the sentence, the corresponding concept of the word is actually determined, and the task here is matching each word obtained by segmentation with the specific concept of the word in HowNet through word disambiguation. Thus, there is no second relationship in the problem of calculating word similarity in HowNet, and the algorithm has become relatively simple. This paper does the multidimensional word sense disambiguation based on HowNet with the segmentation results of LTP platform of Harbin Institute of Technology, then corresponds the word to its corresponding concept, and finally calculates the similarity of concepts. The algorithm of word disambiguation is as follows.

Step 1. Process the target sentence in segmentation and POS (part of speech) tagging with the help of LTP and take all the notional words for calculation.

Step 2. Take out one word for POS disambiguation. Determine the corresponding concept of the word in HowNet according to the POS tagging from word segmentation of LTP. If the corresponding sense to the POS is only one, then the concept of the word can be determined according to the POS tagging. If the corresponding sense to the POS is more than one, then go to Step 3.

Step 3. Example matching disambiguation: take out all the senses of the word with the same POS, put the collocation (including their part of speech) before and after the word to be disambiguated into calculation, and find the corresponding concept in HowNet. The specific steps are as follows:

(1) if the collocation before and after the word to be disambiguated just coincides with the example match of one sense in HowNet, then its concept can be directly determined;

(2) if there is no agreement with the example matching, then do matching calculation according to the POS and the sense of the collocation before and after the word to be disambiguated, and identify the sense with the highest match degree as the concept of the word. For example, the collections to be disambiguated is "da cu (i.e., buy vinegar)"; take "cu (i.e., vinegar)" and all the collections of the verb senses of "da" into matching calculation; then it can confirm that the "da" in "da cu" and the "da" in "da jiang you (i.e., buy soy)" have the same concept.

Step 4. Repeat Step 2, until the concepts of all words are completed.

2.2.2. Adding the Relationship of Antonymy and Oppositeness. Although HowNet provides eight kinds of relationship between sememes in description, but HowNet only uses hyponymy between sememes in the calculation of word similarity. So the range of word similarity value was specified in $[0, 1]$. This paper added antonymy and oppositeness between sememes when computing word similarity on the basis of HowNet; thus the range of word similarity value was extended to $[-1, 1]$. If the two words have the relationship of the following conditions, calculate words similarity according to the following method, else calculate the similarity in accordance with the original method of HowNet:

(1) check converse set and antonym set in HowNet, if the two sememes of their concepts have the relationship of antonymous and converse (the word in italics), for example,

tall: attribute → measurement → stature → *tall*

short: attribute → measurement → stature → *short*

(2) check converse set and antonym set in HowNet, if the two hypernym sememes of their concepts have the relationship of antonymous and converse (the word in italics), for example,

agree: events → static → state → statemental → attitude → *attitudebygood* → agree

disagree: events → static → state → statemental → attitude → *attitudebybad* → disagree

(3) the two sememes of their concepts appear in the sentiment words set of HowNet, such as ⟨positive comment words, negative comment words⟩ and ⟨positive sentiment words, negative sentiment words⟩.

Thus the similarity of the two words is $\mathrm{Sim}(S_1, S_2) = -\mathrm{Sim}(W_1, W_2)$, in which $\mathrm{Sim}(W_1, W_2)$ is the similarity value of the corresponding concepts of the words, namely, the word similarity value.

3. Sentence Similarity Computation

Sentence similarity refers to the matching extent in semantics 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 completely different, its value is 0. Considering the addition of the word relationship of antonymy and oppositeness, this paper expands the value of the sentence similarity to the real interval $[-1, 1]$. The value of -1 indicates that the semantic meaning of the two sentences is completely opposite or completely contrary; the value closer to -1, the bigger the opposite or contrary degree in semantics.

3.1. Algorithm Introduction of Sentence Similarity. At present, the algorithms of the calculation of sentence similarity mainly have three kinds: the method based on keyword information, the method based on semantic information, and the method based on syntactic structure information. Each kind of method with its own features solves the problem of sentence similarity from a different point of view. The specific algorithms at present are the following: string matching method based on keywords [6], the TF-IDF method based on vector space model [7], the calculation method of sentence similarity based on semantic dictionary [3], the calculation method of sentence similarity based on dependency analysis [8], and so on.

The proposing and the practice of these methods have played a very good role in promoting the research on Chinese sentence similarity, but each has its own disadvantages: the method based on keyword information only considers the surface information of the words, without deep understanding; semantic dictionary, despite making up for the defects in the keyword method, is by the limit to the corpus itself and the unknown words; dependency analysis takes into account the sentence dependency structure information, but the algorithm is complex and the cost is too much.

In this paper, with the integration of syntactic structure information, keyword semantic information, and other factors, a calculation method of sentence semantic similarity was put forward based on syntactic structure, in order to improve the accuracy of sentence similarity computation.

3.2. Syntactic Structure. This paper thinks that the overall similarity builds based on the partial similarity. Breaking a complex integration into parts and obtaining the overall similarity through the similarity of the parts helps to solve the problem of sentence similarity computation, in which the key is the syntactic analysis.

A sentence is a language unit of complete meaning that consists of words or phrases according to a certain grammatical structure. The components of a sentence are called the sentence elements; thus words and phrases constitute the sentence elements. Since sentence elements are the units of constructing a sentence, according to the level and the structure characteristics of the sentence, the sentence elements are divided into six kinds: subject, predicate, object, attributive,

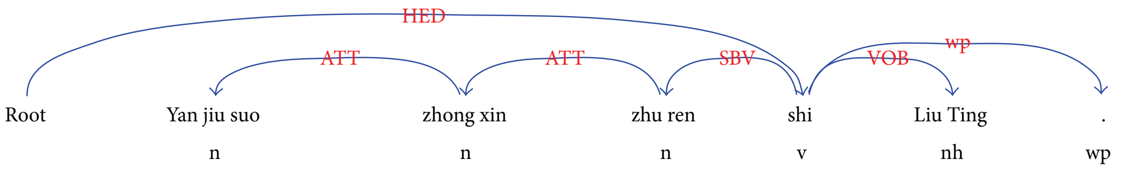
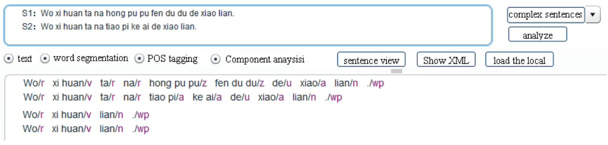


Figure 1: Syntactic analysis example of LTP.


complex sentences ▾
analyze
○ text ○ word segmentation ○ POS tagging ○ Component anaysisi
sentence view
Show XML
load the local

S1: Wo xi huan ta na hong pu pu fen du du de xiao lian.
S2: Wo xi huan ta na tiao pi ke ai de xiao lian.

Wo/r xi huan/v ta/r na/r hong pu pu/z fen du du/z de/u xiao/a lian/n ./wp
Wo/r xi huan/v ta/r na/r tiao pi/a ke ai/a de/u xiao/a lian/n ./wp

Wo/r xi huan/v lian/n ./wp
Wo/r xi huan/v lian/n ./wp

Figure 2: Analysis process of sentence 1 and sentence 2.

adverbial, and complement. Syntactic analysis is to analyze the elements of a sentence and its structure relationship. The method of sentence elements analysis is a common method of syntax analysis, also known as the "central component analysis" [9]. This method thinks that a sentence, regardless of length, contains main component, secondary component, and additional component, of which subject and predicate are the main component, object and complement are the secondary component, and attributive and adverbial are the additional components. The subject which is the statement of the predicate usually precedes the predicate, while the predicate is the core of the whole sentence, so it should first analyze the major components of a sentence, the subject and the predicate, and then determine the other components.

The method of sentence elements analysis rules the corresponding relationship of sentence elements and the words, pays attention to find the corresponding relationship between the sentence elements and the part of speech, and can directly reflect the logical relationship of the meaning the sentence elements express.

Using the method of sentence elements analysis can quickly analyze the trunk and branches of the sentence with more complex structure and contribute to understanding and mastering the sentence. That the structure of the sentence sticks out a mile after sentence elements analysis helps us to compute the sentence similarity based on partial matching.

In this paper, using LTP platform of Harbin Institute of Technology as the sentence analysis tool helps to determine the center words and the sentence elements. LTP is a Chinese natural language processing service platform based on cloud computing technology. LTP has developed an XML-based natural language processing results expression and on this basis provides a rich set of bottom-up, efficient, high-precision Chinese natural language processing modules including lexical, syntactic, semantic analysis, and other five Chinese processing core technologies [1]. A syntactic analysis example of LTP is as shown in Figure 1.

In Figure 1, the "HED," the root points, is the predicate head of the sentence in which its predicate head is "be." In addition, the word segmentation, part of speech tagging, and dependency parsing tagging contained in the analytical structure have provided a good foundation for the calculation of sentence similarity.

3.3. Sentence Similarity Algorithm of This Paper. According to the introduction above, the sentence is divided into three parts by sentence elements analysis: main component, secondary component, and additional component. In order to do the partial component matching calculation better, on the basis of the results of LTP analysis, this paper divided the sentence into three parts: subject component, predicate component, and object component. According to the "HED" LTP tagged, we can determine the predicate component, the subject component on its left, and the object component on its right. From the linguistic knowledge, any sentence is composed by key component (subject, predicate, object, etc.) and modifier component (attributive, adverbial and complement, etc.), while the effect of key component on the sentence is significantly greater than modifier component [10]. Therefore, some articles in the sentence similarity computation only consider the role of key component, but also modifier component of the sentence sometimes has certain influence on similarity computation as follows.

Sentence 1: I like her red and pink face.

Sentence 2: I like his naughty and lovely face.

If only the key component of the sentence is calculated, the component division (including part of speech tagging) can be obtained through the results by LTP.

Component division of sentence 1 is I/r like/v face/n./wp.

Component division of sentence 2 is I/r like/v face/n./wp.

The analysis process is as shown in Figure 2.

In the above sentences, "r" means pronoun, "v" means verb, "n" means noun, "a" means adjective, and "u" means auxiliary.

Thus, the similarity of the two sentences is "1"; that is, sentence 1 and sentence 2 are exactly the same, but the two sentences are different in fact. Therefore, this paper suggests to properly consider the modifier component, in order to improve the accuracy of sentence similarity computation, but some modifiers cannot calculate, such as the adverb in the following.

Sentence 3: He/r jumps/v really/d high/a !/wp.

Component division is He/r jumps high/v !/wp.

Table 1: Part of speech and tagging and abbreviation.

Part of speech	Tagging and abbreviation	Part of speech	Tagging and abbreviation
Noun	n	Pronoun	r
Time noun	t	Adjective	a
Space noun	s	Number	n
Verb	v	Adverb	d

Table 2: Main part of speech of key words in the sentence components.

Key component	The part of speech of key words	Modified component	The part of speech of key words
Subject	n, r	Attributive	n, a, v, r
Predicate	v, a	Adverbial	a, v, s, t, d
Object	n, r	Complement	v, a

After the analysis of annotation results of partially mature corpus (2003 Edition) of "People's Daily" (January, 2000) based on LTP, we found that the subject and the object part of a sentence is mainly a noun or a pronoun, the predicate part is mainly a verb or an adjective, and the part of speech of the words in attributive, adverbial, and complement parts includes more, but the contribution to semantic understanding is also the noun (specifically time nouns and space nouns in adverbial), pronouns, verbs, and adjectives (as Tables 1 and 2 show). Therefore, we will consider all nouns, verbs, pronouns, and adjectives in the sentence in the calculation of sentence similarity and consider two special kinds of part of speech in addition: numerals modifying different quantifiers and adverbs saying "no" (considered in negative sentence). Through word frequency statistics we have also found that the contribution degree of different notional words to the sentence is different. According to the statistical results, nouns, verbs, adjectives, and pronouns were given the weights of 0.3, 0.3, 0.2, and 0.2 (since the frequency of numerals is low, numerals are treated in accordance with pronouns), as a measure of contribution degree to the sentence, and were recorded as θ_1, θ_2, θ_3, and θ_4.

Considering various aspects, this paper puts forward the following calculation formula of sentence semantic similarity:

$$\begin{aligned} \operatorname{Sim}(S_1, S_2) &= \beta_1 \cdot \beta_2 \\ \beta_1 &= k \cdot \lambda \cdot \gamma \cdot \varphi \\ \beta_2 &= \left[\alpha_1 \operatorname{Sim}_1(B_1, B_2) + \alpha_2 \left|\operatorname{Sim}_2(B_1, B_2)\right| \right. \\ &\quad \left. + \alpha_3 \operatorname{Sim}_3(B_1, B_2)\right]. \end{aligned} \tag{3}$$

The value of $\operatorname{Sim}(S_1, S_2)$ is decided by two parts: β_1 which indicates similarity adjustment coefficient and β_2 which represents the semantic similarity value. β_1 also contains four specific parameters: k, λ, γ, and φ.

(1) The parameter k means sentence pattern adjustment coefficient, used to adjust the calculation of sentence similarity in different patterns. Considering the interrogative sentence differs greater in tone with other sentence patterns, the adjustment coefficient in the interrogative sentence relative to imperative sentences, declarative sentences, and exclamatory sentences is set to 0.1; the coefficient between other patterns is set to 0.5; the value of k is 1 in the same sentence pattern.

(2) The parameter λ means sentence component coefficients, namely, the adjustment coefficient when the division of sentence composition is unequal; its value is set to $2 * i/(m+n)$, m and n, respectively, indicate the number of components that S_1 and S_2 contain, and i is the number of corresponding components in S_1 and S_2.

(3) The parameter γ means negative coefficient, while the predicate heads in S_1 and S_2 are obviously antonymous and contrary or before the predicate head in S_1 relative to in S_2 is the word "no", the value of γ is set to -1.

(4) The parameter φ is a supplementary coefficient, namely, if the predicate heads of the two sentences are antonym; besides the subject and the object of the two sentences exchange; then the value of φ is set to -1.

(5) Since a sentence is divided into three parts after syntactic analysis, so the value of β_2 is composed by $\operatorname{Sim}_1(S_1, S_2)$, $\operatorname{Sim}_2(S_1, S_2)$, and $\operatorname{Sim}_3(S_1, S_2)$, respectively, say subject component similarity, predicate component similarity, and object component similarity. Because the value of $\operatorname{Sim}_2(S_1, S_2)$ may be negative, so its absolute value will be taken. α_1, α_2, and α_3 that, respectively, represent the coefficients of the three parts are assigned to the weight of 0.3, 0.5, and 0.2 according to the contribution of the sentence elements and practical experience.

In addition, when the corresponding components of the sentence are compared, there may appear the cases of modifiers with different structure and words with different number (see Table 3 about the following examples); for convenience of calculation, this paper makes the following provisions in the comparison of the same components. If the corresponding components of the two sentences

(1) have the same relationship and more than one word, the center word of the component and other words get

Table 3: Part of experimental data of sentence similarity computation.

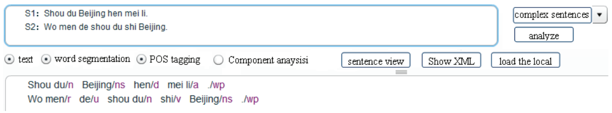
Test sentences	Method 1	Method 2	Method 3
C1: The capital of Beijing is beautiful. (Chinese text: Shou du Beijing hen mei li.) C2: Our capital is Bejing. (Chinese text: Wo men de shou du shi Beijing.)	0.5714	0.5000	0.1276
C3: He knitted a long scarf. (Chinese text: Ta da le yi tiao chang chang de wei jin.) C4: He held up a beautiful umbrella. (Chinese text: Ta da le yi ba piao liang de yu san.)	0.5000	0.6625	0.4274
C5: He went to work today. (Chinese text: Ta jin tian shang ban le.) C6: He didn't go to work yesterday. (Chinese text: Ta zuo tian mei shang ban.)	0.5714	0.5500	−0.7720
C7: I enjoy comfortable life. (Chinese text: Wo xi huan shu shi de sheng huo.) C8: Hands up if you agree. (Chinese text: Tong yi de ren qing ju shou.)	0.0000	0.0000	0.0770
C9: I love to eat apples. (Chinese text: Wo ai chi ping guo.) C10: The fruit I like are apples. (Chinese text: Wo xi huan de shui guo shi ping guo.)	0.4444	0.3600	0.1928
C11: How is she recently? (Chinese text: Ta zui jin hao ma?) C12: She is very fine recently. (Chinese text: Ta zui jin hen hao.)	1.0000	0.7750	0.1000
C13: He pushed his younger brother over. (Chinese text: Ta ba di di tui dao le.) C14: His younger brother was pushed over by him. (Chinese text: Di di bei ta tui dao le.)	0.7500	0.5400	1.0000

the weight in accordance with the distribution of 0.6 and 0.4, such as ADV (adverbial structure) in C5 and C6;

(2) have the same relationship and more than one noncentral word, the POS of noncentral words gets the weight according to the proportion of noun, verb, adjective, and pronoun. If the POS of more than one word are the same, calculate their similarity, respectively, taking the maximum;

(3) have part of the same relationship, only compare the same relationship, such as ATT (attributive structure) in one component, ATT (attributive structure) and VOB (verb-object structure) in another component, finally coordinate with granularity coefficient which is $2 * i/(m+n)$, in which i is the number of the same relationship within the corresponding components, and m and n, respectively, represent the number of relationship in the corresponding components of the two sentences;

(4) have different relationship, only calculate the similarity for the parts with the same part of speech and finally coordinate by granularity coefficient which is $2 * i/(m+n)$, i is the number of words with the same part of speech, and m and n, respectively, represent the number of words in the corresponding components of the two sentences, such as the object component in C9 and C10; if the relationship does not match, do the comparison of the center word, such as the subject component in C7 and C8.


S1: Shou du Beijing hen mei li.
S2: Wo men de shou du shi Beijing.

Shou du/n Beijing/ns hen/d mei li/a ./wp
Wo men/r de/u shou du/n shi/v Beijing/ns ./wp

Figure 3: Syntactic analysis result of LTP.

Besides, if the two sentences, respectively, contain the word "Ba" and the word "Bei" and the similarity between the subject (object) of a sentence and the object (subject) of another sentence is greater than a certain threshold (0.5), then the similarity of two sentences is 1.

Here is the first set of examples in the experiments, which shows the calculation process of sentence similarity:

(1) syntactic analysis (as shown in Figure 3),

(2) component analysis (as shown in Figure 4),

(3) word disambiguation:

S_1: capital/n DEF = {place: PlaceSect = {capital}, belong = {place: PlaceSect = {country}, domain = {politics}}}

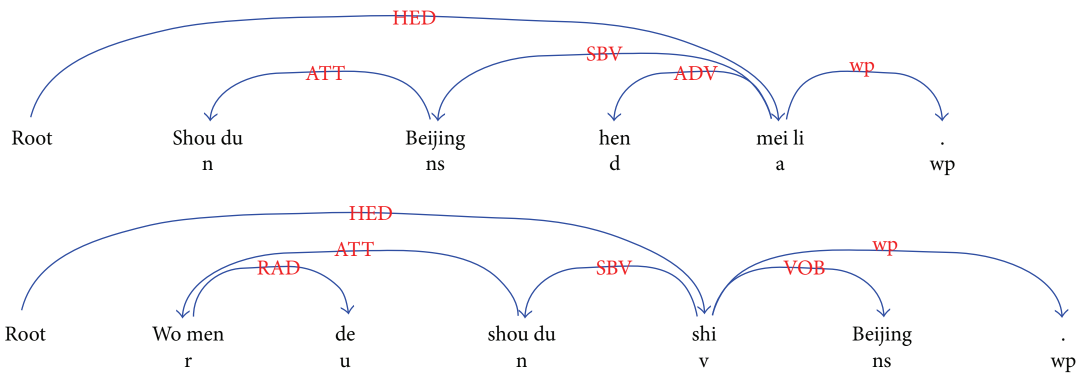


Figure 4: Component analysis result of LTP.

Beijing/n DEF = {place: PlaceSect = {capital}, belong = “China”, modifier = {ProperName}}

beautiful/a DEF = {beautiful}

S_2: we/r DEF = {specific: PersonPro={1st Person}, quantity = {mass}}

capital/n DEF = {place: PlaceSect = {capital}, belong = {place: PlaceSect = {country}, domain = {politics}}}

is/v DEF = {be}

Beijing/n DEF = {place: PlaceSect = {capital}, belong = “China”, modifier = {ProperName}}

(4) sentence similarity computation:

$$\begin{aligned}\operatorname{Sim}(S_1, S_2) &= \frac{2 * i}{m+n} \times \{0.3\operatorname{Sim}[(\text{capital, Beijing}), (\text{we, capital})] \\ &\qquad + 0.5\operatorname{Sim}[(\text{beautiful}), (\text{is})]\} \\ &= \frac{2 \times 2}{2+3}[0.3 \times (0.4 \times 0.0444 + 0.6 \times 0.7333) \\ &\qquad + 0.5 \times 0.0444] = 0.1276.\end{aligned} \tag{4}$$

3.4. Results and Analysis of the Experiment. This paper selects three kinds of methods of sentence similarity to make comparisons.

Method 1. Methods based on the keyword, such as the sentence similarity computing formula proposed by Peking University Institute of Computational Linguistics [11], are as follows:

$$\frac{2c}{m+n}, \tag{5}$$

wherein c represents the number of key words appearing together in the two sentences and m and n represent the number of key words in the two respective sentences.

Method 2. The sentence similarity in the methods based on the combination of the word form and word order depends on the surface similarity and word order similarity, please see the literature of teacher He and Wang [12], the formula is

$$\operatorname{Sim}(S_1, S_2) = \alpha_1 \operatorname{Sim}_s(S_1, S_2) + \alpha_2 \operatorname{Sim}_p(S_1, S_2), \tag{6}$$

wherein $\operatorname{Sim}_s(S_1, S_2)$ represents surface similarity, $\operatorname{Sim}_p(S_1, S_2)$ represents the word order similarity, α_1 and α_2 are constants, and $\alpha_1 + \alpha_2 = 1$.

Method 3. The similarity calculation method based on syntactic structure in this paper, please refer to formula (3).

Some experimental data of sentence similarity computation are shown in Table 3.

It is seen from the results of the experiment in Table 3 that Methods 1 and 2 cannot reflect the essential relations between sentences well; the algorithm in this paper considers the semantic relations between sentences better from the perspective of semantics, and the results are more suitable with the intuitive understanding of people, such as the comparison of C3 and C4; this algorithm also considers the influence of the sentence pattern which conveys semantics to the sentence, adds the sentence pattern adjustment coefficient, and makes the results more in line with the understanding of people than the other two methods, such as C11 and C12; sentence similarity computation with the relationship of antonymy or oppositeness is also designed in this paper (such as C5 and C6), which makes the calculation results of positive and negative distinction; the closer the value to −1, the closer the meaning of the two sentences to antonymy or oppositeness; in addition, the algorithm was designed between declarative sentences, “Ba” sentences and “Bei” sentences from syntactic structure, which makes the similarity calculation more accurate, such as C13 and C14. But the structure of Chinese sentence is complex and it is not suitable for all sentences using unified algorithm, such as C9 and C10; the result is 0.1928 with our algorithm; obviously this similarity value cannot reflect the original intention of the two sentences. Here Shi-sentence is involved, in which the subject and the object represent the same semantic relationship in one sentence, and the subject and object can be interchangeable, so we will consider appropriate conversion of sentence structure in the future; of course there is a lot of problems that need to be solved.

4. Conclusion

Based on the analysis of the existing sentence similarity algorithm, the sentence semantic similarity algorithm based on syntactic structure was proposed, and adding semantic similarity computation in the premise of syntactic similarity can greatly improve the effectiveness of retrieval. First of all, do an analysis on sentence components with the help of the LTP platform of Harbin Institute of Technology; then make semantic comparisons based on syntactic structure of the sentence. Since the meaning of a sentence depends ultimately on the meaning of the words, in this paper, the word similarity was calculated after the word was translated into the concept through word disambiguation with the help of HowNet, hereby realizing semantic comprehension. Meanwhile considering modifier words in the sentence components with certain contribution, it gives more detailed rules to further distinguish the relations between sentences. The experimental results show that, under the same test condition, the proposed algorithm results are more in accordance with the actual situation and obviously improve the calculation accuracy. By the limit of LTP platform of Harbin Institute of Technology and HowNet, such as the imperfect definition of the word concepts in HowNet and the deviation in POS tagging of the two platforms, to a certain extent affected the accuracy of sentence similarity computation. In addition, we will also go to deep research on complex sentence structures to solve the problem of special sentence patterns such as "be" sentences.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Thanks are due to National Natural Science Foundation of China (60973051) and Science and Technology Department of Henan Province Key Scientific and Technological Project (112102210375).

References

[1] J. Lang, T. Liu, and S. Li, "LTP: an XML-based open language technology platform," in *Proceedings of the 25th Annual Academic Conference of Chinese Information Processing Society of China*, 2006.

[2] Z. D. Dong and Q. Dong, "HowNet 1999[EB/OL]," 2014, http://www.keenage.com.

[3] 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.

[4] Z. X. Zhang and J. H. Li, "A word, similarity computing method based on superposed degree of sememe," *Journal of Xinyang Nomal University*, vol. 23, no. 2, pp. 296–299, 2010.

[5] D. Lin, "An information theoretic definition of similarity," in *Proceedings of the 15th International Conference on Machine Learning (ICML '98)*, pp. 296–304, 1998.

[6] X. Q. Lu, F. L. Ren, and Z. D. Huang, "Sentence similarity model and the most similar sentence search algorithm," *Journal of Northeastern University*, vol. 24, no. 6, pp. 531–534, 2003.

[7] Y. Wang, B. Qin, and S. F. Zheng, "Sentence similarity for frequently-asked question," in *Proceedings of the 1st workshop on Computational Linguistics*, 2002.

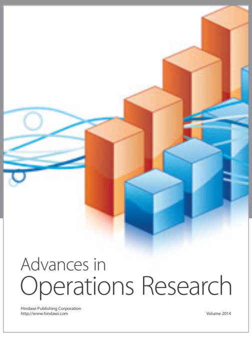
[8] B. Li, T. Liu, and B. Qin, "Chinese sentence similarity computing based on semantic dependency relationship analsis," *Applications Research of Computers*, vol. 20, no. 12, pp. 15–17, 2003.

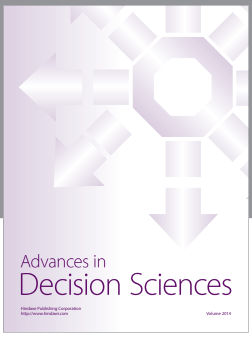
[9] "The method of sentence constituent analysis[EB/OL]," 2014, http://baike.baidu.com/view/338602.htm?fr=aladdin.

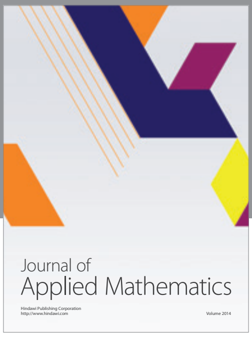
[10] J. Li, *The Research on Block-Based Semantic Similarity of Sentences*, Anhui University of Technology, Hefei, China, 2011.

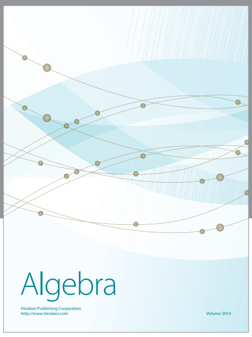
[11] C. P. Cheng and Z. G. Wu, "A method of sentence similarity computing based on HowNet," *Computer Engineering & Science*, vol. 32, no. 2, pp. 172–175, 2012.

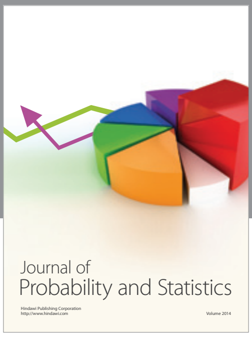
[12] W. He and Y. Wang, "Text representation based on sentence and Chinese text categorization," *Journal of the China Society for Scientific and Technical Information*, vol. 28, no. 6, pp. 839–843, 2009.


Advances in Operations Research

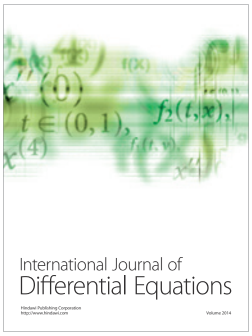

Advances in Decision Sciences


Journal of Applied Mathematics

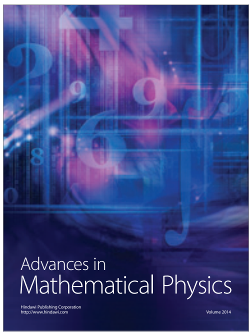

Algebra

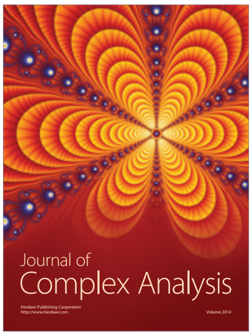

Journal of Probability and Statistics

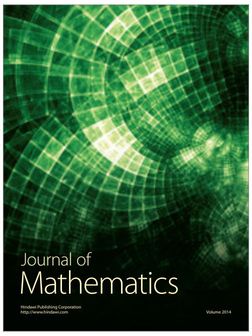
The Scientific World Journal

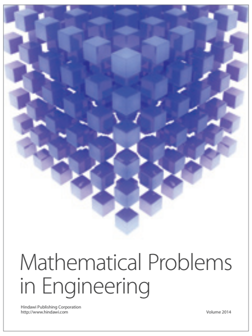

International Journal of Differential Equations

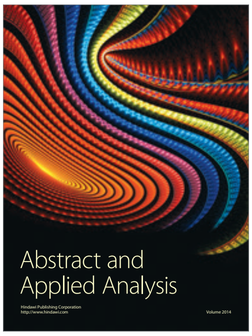
International Journal of Combinatorics

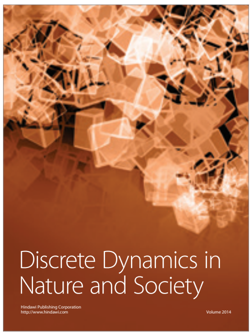

Advances in Mathematical Physics

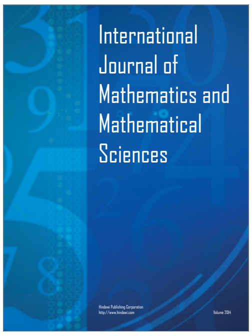

Journal of Complex Analysis


Journal of Mathematics


Mathematical Problems in Engineering

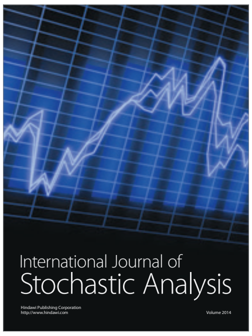

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


International Journal of Stochastic Analysis

Journal of Optimization