Word Similarity Computing Based on Hybrid Hierarchical Structure by HowNet*

JINAN XU, JIANGMING LIU AND YUJIE ZHANG
School of Computer Information and Technology
Beijing Jiaotong University
Beijing, 100044 P.R. China
E-mail: {xja2010; jmliunlp}@gmail.com; yjzhang@bjtu.edu.cn

Word similarity computing is one of the most important and fundamental tasks in the field of natural language processing. Most of word similarity methods perform well in synonyms, but not well between words whose similarity is vague. It confronts the challenge of how to overcome this problem. An approach is proposed to compute Chinese word similarity based on hybrid hierarchical structure by HowNet to achieve fine-grained similarity results. The experimental results prove that the method has a better effect on computing similarity of synonyms and antonyms including nouns, verbs and adjectives. In addition, it performs well and stably on standard data provided by SemEval 2012.

Keywords: word similarity computing, hybrid hierarchical structure, HowNet, concept, WordNet

1. INTRODUCTION

Word similarity computing plays an important role in various fields, for instance Natural Language Understanding and Cognitive Science. There are many applications requiring word similarity like Information Retrieve, Machine Translation, Sentiment Analysis [1, 2], Question Answer systems [3, 4], and Opinion Mining [5]. Moreover, it is a pivotal method in Word Sense Disambiguation (WSD).

Two main types of word similarity computing methods have been proposed. One usually using thesaurus is based on the knowledge of semantics or classification. Thesaurus is used to organize all words under a certain structure such as hierarchy. The methods of this type sufficiently utilize the structure of thesaurus, such as the path between two words in net [6] and depth in semantic trees [7]. They have the advantages of preciseness and deep usage of word semantics, but a relatively complete semantic dictionary is required in order to ensure the presence of words in thesaurus. The other type is based on large-scale corpus. These methods of this type use large-scale corpus to compute words similarity. But they have some inevitable disadvantages, such as the need of large-scale corpus, noise, low search efficiency etc. [8]. Therefore, it is fine to create a thesaurus with Internet resource or large-scale corpus [9-11] as an interim for computing word similarity. So methods based on thesaurus are necessary.

WordNet is deemed to be very valuable resource in computing similarity between English words. Since Chinese that belongs to isolated language is different from English that belongs to inflected language and the complex Chinese grammar is highly ambiguous, computing Chinese words similarity is more difficult than English under the same

Received October 8, 2014; accepted September 30, 2014.
Communicated by Hsin-Min Wang.
* Project 61370130 supported by National Natural Science Foundation of China.
lack of systematic resource. HowNet is a valuable bilingual knowledge resource organized by Zhongdong Dong.


In this paper, the accuracy and stability have been significantly improved taking the DEF definition of sememe in hierarchic structure into consideration. The proposed algorithm by this paper fuses hierarchic DEF definition of sememe and hierarchic structure of sememe. It performs better and stably no matter between the high similarity words namely synonyms and between the vague similarity words.

The remainder of the paper is organized as follows: Section 2 briefly introduces HowNet and relative knowledge. Section 3 describes our algorithm in details. Section 4 presents the experimental results and comparison. In the last section, conclusion is put forward and future work is discussed.

2. HOWNET

HowNet is reasonable and adaptable, whose semantic description is rich, and the case of one word description of HowNet is shown in Fig. 1. HowNet uses a markup language called KDML to describe word’s concept which facilitates computer processing [13]. There are more than 173 thousand of words in HowNet which are described by bilingual DEF. A different semantic of one word has a different DEF description. DEF is defined by a number of sememes and the descriptions of semantic relations between words. It is worth to mention that sememe is the most basic and the smallest unit which cannot be easily divided [6], and the sememes are extracted from about six thousand of Chinese characters [12].

An example of one DEF of fishing pole can be described as follows:

\[
\text{DEF} = \{\text{tool|用具}: \{\text{catch|捉住}: \{\text{instrument|~}, \text{patient|鱼}\}\}\}.
\]

Among the above example, the words describing DEF, such as tool, catch, instru-
ment, patient and fish, are sememes. Then the description of DEF is a tree-like structure as shown in Fig. 2. The relation between different sememes in DEF is described as a tree structure in taxonomy document of HowNet. In taxonomy event and taxonomy entity, sememes are also described by DEF which is the detailed description of sememe tree in HowNet 2008. Therefore, in HowNet, words that are described by DEF and sememes are also specifically described by DEF which is shown in Fig. 3.

3. SIMILARITY COMPUTING

3.1 DEF Similarity Computing

In Fig. 2, the hierarchy of DEF is clearly introduced. DEF is tree-like structure. Tree-like structure similarity comes from node similarity in tree. Due to different relation on the edge of tree, DEF similarity computing is not conventional tree similarity computing. This is one of our core works.

In DEF similarity, the node similarity between one pair of nodes in the same layer of tree comes from two types of similarity, namely the relation similarity from that of its children nodes and sememe similarity itself which is described later in detail in section 3.2.
For relation similarity, we take saleslady (Fig. 4) and conductor (Fig. 5) for example [12] which is similar on morphology. When computing a pair of node similarity, such as root nodes, which are regarded as current calculating nodes (CN). Then both of CN themselves and the children nodes of CN are taken into consideration. CN (human) of saleslady has relations of hostof, domain, modify and none (no relation). With the same relations in CN (human) of conductor, we get the similarity of children nodes as one relation similarity of a pair of CN. In other words, the similarity of children nodes which have the same relation with their respective father nodes will be computed. If there is no match, the relation similarity is defaulted as small constant $\delta$. Every pair of nodes should be calculated in DEF tree in same layer as formula (1).

$$Sim_{rela}(S_1, S_2) = \beta_{rela} \frac{1}{N} \sum_{i=1}^{N} Sim_{rela,i}(S_1, S_2) + \beta_{s} Sim_{s}(S_1, S_2).$$

(1)

Where, $N$ denotes $N$ different kinds of relation, $Sim_{rela}(S_1, S_2)$ denotes the $i$th relation similarity which in fact expresses the children node similarity of the pair $(S_1, S_2)$, $Sim_{s}(S_1, S_2)$ denotes sememe similarity, and $\beta_{rela}$$\geq 0$, $\beta_{s}$$\geq 0$, $\beta_{rela}+\beta_{s}=1$. Bottom-to-up algorithm will be used to recursively compute DEF similarity in order to achieve the root node similarity as the DEF similarity. The formula is below.

$$Sim_{DEF}(S_1, S_2) = Sim_{node}(S_1, S_2) \text{ if } S_1 = \text{root1, } S_2 = \text{root2}. $$

(2)

The key point of DEF similarity computing method is not only taking the migration process of the nodes in the DEF tree into consideration [12], but also using the relation between children nodes and their respective father node. Through this method, the structure information of DEF tree can be deeply used.

When computing sememe similarity $Sim_{s}(S_1, S_2)$, if sememe belongs to Attribute Sememe and Secondary Feature Sememe, its weight is so high that the similarity unreasonably increases. Take “Hardness” and “Reputation” as examples.
DEF = \{\text{Hardness|硬度.host} = \{\text{physical|物质}\}\}
DEF = \{\text{Reputation|名声.host} = \{\text{group|群体}\{\text{human|人}\}\}\}.

The two sememes in Attribute sememe tree so closely represent the words’ attributes that DEF similarity reaches high (0.6423). So we add a penalty factor $\varepsilon$ to reduce the noise produced by Attribute Sememe and Secondary Feature Sememe. Therefore, if a pair of sememes both belongs to Attribute Sememe or Secondary Feature Sememe, the formula (2) derived from the formula (1) is used to compute node similarity.

$$\text{Sim}(S_1, S_2) = \frac{1}{N} \sum_{i=1}^{N} \text{Sim}_{rela}(S_1, S_2) + \varepsilon \beta \text{Sim}_{meta}(S_1, S_2).$$

(3)

3.2 Sememe Similarity Computing

Section 3.1 discusses how to calculate the similarity of DEF. In this section, we discuss how to calculate the sememe similarity $\text{Sim}_s(S_1, S_2)$ in DEF similarity. As Fig. 3 shows, there exists a path between sememes in hierarchy, and meanwhile the sememes are also described by DEF. Therefore, sememe similarity can be divided into two parts, namely DEF similarity and structure similarity. Most of researchers only consider structure similarity without DEF similarity in sememe tree.

3.2.1 Structure similarity between sememes

Liu [6] proposes a method by the distance of two sememes in sememe tree. Li [17] considers the depth in sememe tree structure. Ge Bin [7] puts forward density in sememe tree structure and uses the least common nodes (LCN) to compute the sememe similarity. It will be more reasonable and finer to compute similarity of words with the same distances, combing density and depth. This paper uses formula (3) below to compute structure similarity of sememe similarity.

$$\text{StructSim}(S_1, S_2) = \frac{\alpha \cdot (\text{depth}(S_1) + \text{depth}(S_2))}{\alpha \cdot (\text{depth}(S_1) + \text{depth}(S_2)) + \text{dist}(S_1, S_2) + \left| \text{depth}(S_1) - \text{depth}(S_2) \right|}$$

(4)

Where, $\text{depth}(S_i)$ represents depth of $S_i$ in sememe tree, and $\text{dist}(S_1, S_2)$ is the distance between $S_1$ and $S_2$ in sememe tree. Obviously, structure similarity of sememes will increase along with the decreasing of the distance between sememes and the closer depth of two sememes.

Besides, the antonym relation and converse relation are important. In process of computing structure similarity between $S_1$ and $S_2$, the following extra steps are considered to ensure that the antonym and converse relationship is never ignored.

- If there exists a antonym relation or converse relation between $S_1$ and $S_2$, or so does the same relation in the path between $S_1$ and $S_2$ in sememe tree, mark a flag with “-”;
- Due to the description of some DEF containing real words not sememes in HowNet, the similarity between real words is 1 if real words are same, otherwise is 0. The similarity between one real word and sememe is defaulted as small constant $\gamma$. 


However, antonym relation or converse relation listed in HowNet document is too strict. For example, “unsightly|难看” and “GoodLooking|好看” that constitute a pair of antonym and “ugly|丑” and “beautiful|美” make up another pair of antonym, yet in HowNet, “beautiful|美” and “unsightly|难看” are not used as a pair of antonym. So Synonym Dictionary is used to find synonym relation between “GoodLooking|好看” and “beautiful|美” in order to extend antonym and converse relation, namely “beautiful|美” and “unsightly|难看”.

3.2.2 DEF similarity between sememes

Section 3.1 has discussed DEF similarity in detail. The method of DEF similarity computing of sememes is the same as the method of DEF similarity computing of words. The special phenomenon can be found in two aspects. On one hand, in process of computing DEF similarity, Sememe similarity computing is need. On the other hand, in process of computing sememe similarity, DEF similarity computing is need. This phenomenon brings about a cyclical calculation. In experiments, in order to terminate infinite cyclical calculation, cyclical calculation will be processed only twice. In process of computing sememe similarity in the last calculation circle, we only use structure similarity as sememe similarity. So the more times cyclically calculation, the more detailed computing similarity is, and finally result will be convergent. The definition of convergence will be researched in our future work.

Formula (4) below is given for combining DEF similarity with structure similarity to get sememe similarity.

\[ \text{Sim}_{\text{DEF}}(S_1, S_2) = \begin{cases} \text{StructSim}(S_1, S_2) & \text{if last circle} \\ \beta_{\text{struct}} \cdot \text{StructSim}(S_1, S_2) + \beta_{\text{DEF}} \cdot \text{Sim}_{\text{DEF}}(S_1, S_2) & \text{if not last circle} \end{cases} \]  

Where, \( \text{StructSim}(S_1, S_2) \) denotes structure similarity. \( \text{Sim}_{\text{DEF}}(S_1, S_2) \) is the DEF similarity, and \( \beta_{\text{struct}} \geq 0, \beta_{\text{DEF}} \geq 0 \), \( \beta_{\text{struct}} + \beta_{\text{DEF}} = 1 \). \( \text{Sim}_{\text{DEF}}(S_1, S_2) \) equals 1 if there is no DEF description of sememe in both \( S_1 \) and \( S_2 \).

3.2.3 DEF similarity between sememes

In sections 3.1 and 3.2, we specifically discuss the method to compute DEF similarity and sememe similarity and the method to combine them. In HowNet, a word has one or more DEF descriptions in different semantics. Formula (5) below will be used to compute similarity between words by

\[ \text{Sim}_{\text{DEF}}(W_1, W_2) = \max_{i, j} \left| \text{Sim}_{\text{DEF}}(S_{i1}, S_{j1}) \right| \]  

Where, \( S_{i1} \) is the \( i \)th DEF of word \( W_1 \), \( S_{j1} \) is the \( j \)th DEF of word \( W_2 \), “+” or “-” depends on the flag (section 3.2) of max DEF similarity, and \( \text{Sim}_{\text{DEF}} \) means DEF similarity. In formula (5), we choose maximum DEF similarity as word similarity by default. Furthermore, according to the word context, choosing a suitable DEF for a word is necessary, which will be introduced in our future papers.
4. EXPERIMENTS AND ANALYSIS

Based on HowNet and the algorithm described above, we implement a program to compute word similarity [18]. General parameters in experiments derive from Liu’s [6] and Li’s [13]. The special parameters $\beta_{\text{struct}}$ and $\beta_{\text{DEF}}$ are optimized on the datasets of Evaluating Chinese Word Similarity task In SemEval 2012 [14] to minimize difference of similarity between experimental results and golden results. Through greedy algorithm of parameter estimation, $\beta_{\text{struct}}$ is set 0.7 and $\beta_{\text{DEF}}$ is set 0.3. Table 1 gives all the parameters of experiments.

<table>
<thead>
<tr>
<th>General parameter</th>
<th>$\alpha$</th>
<th>$\delta$</th>
<th>$\gamma$</th>
<th>$\beta_{\text{rel}}$</th>
<th>$\beta_{\varepsilon}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1.6</td>
<td>0.1</td>
<td>0.1</td>
<td>0.3</td>
<td>0.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Special parameter</th>
<th>$\beta_{\text{struct}}$</th>
<th>$\beta_{\text{DEF}}$</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0.4</td>
<td>0.6</td>
<td>0.1</td>
</tr>
</tbody>
</table>

The value of our result of experiment can be divided two parts, namely degree of similarity and antonymous symbol. The larger absolute value of result, the higher similarity words have. If the symbol of result is “-”, there exists antonym or converse relation.

4.1 Nouns and Verbs Experiments

A group of words are chosen from Li’s paper and Liu’s paper as examples [6, 13], and show the result of our approach contrasted with Liu’s and Li’s in Table 2 [6, 13]. In Liu’s paper, the similarity between words, such as pair of “man” and “father” and pair of “pink” and “crimson”, is unreasonable [13]. Our algorithm performs as well as Li’s in solving this problem. What’s more, through adding flag to mark antonym relation, our algorithm performs better than Li on some pairs of words, such as “man” and “woman”. “man” and “woman” is a pair of absolute antonyms, so our method gets higher similarity with a flag (-) marking antonym.

<table>
<thead>
<tr>
<th>Word 1</th>
<th>Word 2</th>
<th>Liu’s result [6]</th>
<th>Li’s result [13]</th>
<th>Our result</th>
</tr>
</thead>
<tbody>
<tr>
<td>男人 (man)</td>
<td>女人 (woman)</td>
<td>0.8611</td>
<td>0.8955</td>
<td>−0.9957</td>
</tr>
<tr>
<td>男人 (man)</td>
<td>父亲 (father)</td>
<td>1.0000</td>
<td>0.8902</td>
<td>0.8904</td>
</tr>
<tr>
<td>男人 (man)</td>
<td>母亲 (mother)</td>
<td>0.8611</td>
<td>0.7857</td>
<td>−0.8875</td>
</tr>
<tr>
<td>男人 (man)</td>
<td>和尚 (monk)</td>
<td>0.8611</td>
<td>0.4981</td>
<td>0.5170</td>
</tr>
<tr>
<td>男人 (man)</td>
<td>经理 (manager)</td>
<td>0.6303</td>
<td>0.5116</td>
<td>0.5200</td>
</tr>
<tr>
<td>男人 (man)</td>
<td>高兴 (happy)</td>
<td>0.0741</td>
<td>0.0002</td>
<td>0.0100</td>
</tr>
</tbody>
</table>
Table 2. (Cont’d) Comparison of nouns and verbs.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>男人 (man)</td>
<td>收音机 (radio)</td>
<td>0.1078</td>
<td>0.2133</td>
</tr>
<tr>
<td>粉红色 (pink)</td>
<td>深红色 (crimson)</td>
<td>1.0000</td>
<td>0.8500</td>
</tr>
<tr>
<td>风度 (comportment)</td>
<td>面积 (square measure)</td>
<td>0.6131</td>
<td>/</td>
</tr>
<tr>
<td>名声 (reputation)</td>
<td>硬度 (hardness)</td>
<td>0.6176</td>
<td>/</td>
</tr>
<tr>
<td>发明 (invent)</td>
<td>创造 (create)</td>
<td>0.6153</td>
<td>/</td>
</tr>
<tr>
<td>三伏 (hot)</td>
<td>冬眠 (hibernate)</td>
<td>0.0429</td>
<td>/</td>
</tr>
</tbody>
</table>

4.2 Adjectives Experiments

A group of adjectives are chosen from emotion words as examples. Li’s algorithm and Liu’s algorithm [6, 13] never take antonym relation into consideration in computing word similarity. Jiang [15] extends Liu’s algorithm by using antonym relation. Therefore, our results are only contrasted with Jiang’s [15] shown in Table 3. Table 3 shows that our result is much better than Jiang’s [15] in many words, such as “beautiful” and “shifty-eyed”. As we know, “beautiful” and “shifty-eyed” strictly is a pair of antonyms, and “shifty-eyed” is “ugly” but not vice versa. So value of similarity between “beautiful” and “shifty-eyed” is unreasonable. For another example of “comfortably” and “handicap”, our approach performs better.

Table 3. Comparison of adjectives.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Jiang’s result [15]</th>
<th>Our result</th>
</tr>
</thead>
<tbody>
<tr>
<td>美丽 (beautiful)</td>
<td>丑陋 (ugly)</td>
<td>−1.0000</td>
<td>−1.0000</td>
</tr>
<tr>
<td>美丽 (beautiful)</td>
<td>贼眉鼠眼 (shifty-eyed)</td>
<td>−1.0000</td>
<td>−0.9662</td>
</tr>
<tr>
<td>美丽 (beautiful)</td>
<td>优雅 (elegant)</td>
<td>0.7884</td>
<td>0.9264</td>
</tr>
<tr>
<td>高尚 (exalted)</td>
<td>卑鄙 (base)</td>
<td>−0.9125</td>
<td>−0.9364</td>
</tr>
<tr>
<td>舒服 (comfortably)</td>
<td>疼痛 (handicap)</td>
<td>−0.0664</td>
<td>−0.7989</td>
</tr>
<tr>
<td>勇敢 (brave)</td>
<td>坚强 (strong)</td>
<td>0.7884</td>
<td>0.9500</td>
</tr>
</tbody>
</table>

4.3 Synonyms Experiments

We use the synonym groups from the extended edition of Tong Yi Ci Lin and compute similarity among them. The extended edition of Tong Yi Ci Lin covers 77 thousand of words. There exist some words in the extended edition of Tong Yi Ci Lin that are not appearing in HowNet. So we randomly choose nearly 8000 pairs of words,
which are both in the extended edition of Tong Yi Ci Lin and HowNet, as data of experiments. The result of synonyms experiment is shown in Fig. 6.

It illustrates the effectiveness of our approach since most of similarity of synonyms is very high. However, there is a small number of pair of synonyms with low similarity computed through our approach. This phenomenon is caused by definitions in HowNet which is different from the extended edition of Tong Yi Ci Lin. Table 4 shows that our approach performs better than Li’s in computing similarity of synonyms.

Table 4. Percentage of synonyms in different ranges.

<table>
<thead>
<tr>
<th>approach</th>
<th>Similarity(&gt;0.9)</th>
<th>Similarity(&gt;0.8)</th>
<th>Similarity(&gt;0.7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li’s approach</td>
<td>60.89%</td>
<td>68.75%</td>
<td>72.85%</td>
</tr>
<tr>
<td>Our approach</td>
<td>66.05%</td>
<td>70.21%</td>
<td>74.76%</td>
</tr>
</tbody>
</table>

4.4 Antonyms Experiments

Due to a lack of regular antonym resource, all the antonyms comes from web resource using crawler and manually select correct antonyms to reduce data noise produced by web. Then nearly 3000 pairs of antonyms appearing in HowNet is selected, including 475 pairs of antonyms where each word has only one Chinese character, 2200 pairs of antonyms where each word has two Chinese characters, and 325 pairs of antonyms where each word has four Chinese characters.

The similarity of antonyms comes from two parts. One is the value of result of antonyms experiment denoting antonymous degree that is shown in Fig. 7, and the other is the flag “-” (section 3.2) marking antonym. Table 5 shows the percent of antonym in different ranges of similarity illustrating our approach performance in computing antonyms similarity. Table 6 shows the number of pairs of antonyms with flag “-” by our approach.

Through experiments on computing similarity of antonyms, it proves the high effectiveness of our approach of computing word similarity for most of similarity of antonyms. However, it performs not very well in finding the flag “-” (section 3.2) which marks antonym. One of the main reasons is that antonyms and converse sememe description in HowNet are not deeply complete. With the development of HowNet, our approach will perform better.
Table 5. Percentage of synonyms in different ranges.

<table>
<thead>
<tr>
<th>similarity</th>
<th>percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;0.9</td>
<td>50.02%</td>
</tr>
<tr>
<td>&gt;0.8</td>
<td>61.89%</td>
</tr>
<tr>
<td>&gt;0.7</td>
<td>68.70%</td>
</tr>
</tbody>
</table>

Table 6. Number of antonyms with flag “-”.

<table>
<thead>
<tr>
<th>method</th>
<th>number in 3000 pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>863</td>
</tr>
<tr>
<td>extend</td>
<td>966</td>
</tr>
</tbody>
</table>

4.5 SemEval Experiments

In SemEval, the datasets of Evaluating Chinese Word Similarity task in SemEval 2012 is used as the experimental data, and the value of the result of manual annotation is normalized as [0, 1]. The experimental data which exclude words that do not exist in HowNet remains 130 pair words, including 40 pairs of words whose similarity between 1 and 0.6, 50 pairs of words whose similarity between 0.6 and 0.3, and 40 pairs of words whose similarity between 0.3 and 0. Experimental data is sequenced by their similarity from high to low. Experiments use datasets from 20 to 100 by the 20 step size and 130 (the whole). The result of experiments is shown in Fig. 8.

In Fig. 8, the value of vertical axis means the difference similarity value between the experiment results and standard results, and the value of horizontal axis is the size of set. Compared with Liu’s method [6], the result shows that in the high similarity pairs of words, the difference of similarity is nearly 0.095. Besides, the largest difference is lower than 0.1 which can be accepted. In Fig. 8, it shows that the difference value between the highest difference and lowest difference is 0.01. It is verified that the approach proposed by this paper is effective and stable in different range of similarity.

Due to the sum of $\beta_{struct}$ and $\beta_{DEF}$ equals to one. Therefore, if $\beta_{DEF}$ is fixed, $\beta_{struct}$ equals to 1-$\beta_{DEF}$. So results of the experiment on influence of Parameter $\beta_{DEF}$ is shown in Fig. 9, where the value of $\beta_{DEF}$ is from 0.1 to 0.9 by 0.1 step size. Fig. 9 shows that value of $\beta_{DEF}$ highly influence the result of similarity computing.
A new approach for computing word similarity between Chinese words using HowNet is proposed. The results of our experiments verify the effectiveness of our approach. Our approach is compared with the Liu’s [6] approach, Li’s approach [13] and Jiang’s [15] approach in different experiments, such as nouns and verbs, adjectives, synonyms, antonyms and SemEval 2012. We conclude our works in three aspects. Our approach is effective to precisely compute word similarity, and improves the accuracy of similarity by using DEF description in sememe hierarchy. In one concept, different kinds of sememe describe DEF in different weight. The Synonym Dictionary can be used to alleviate strict limitations in antonym and converse relation, which assists our approach performing better.

Due to the importance of word context, in future, in document, the method to choose suitable DEF for the word is necessary depending on context instead of maximum DEF similarity calculated by formula (5). Moreover, In the process of computing DEF similarity, the sub-tree constraint may be also important. Some DEF descriptions are the sub-description of other DEF description, so the alignment between sub-description of DEF is meaningful in computing semantic similarity. We will pay more attention to the information of alignment of sub-tree. In the next work, we will expand the standard dataset for experiments to optimize parameters for various applications.
REFERENCES


**JinAn Xu (徐金安)** is currently an Associate Professor in School of Computer Science and Information Technology, Beijing Jiaotong University, Beijing, China. He received the B.Eng. degree from Beijing Jiaotong University in 1992, the Master degree and Ph.D. degree in Computer Information from Hokkaido University, Sapporo, Japan, in 2003 and 2006, respectively. His research focuses on natural language processing, machine translation, information retrieve, text mining, and machine learning.

**JiangMing Liu (刘江鸣)** is a MS student in School of Computer Information and Technology at Beijing Jiaotong University, China. His research focuses on summarization and semantic analysis for machine translation.

**Yujie Zhang (张玉洁)** is Professor of Computer and Information Technology at Beijing Jiaotong University. She received the B.Eng. degree from Beijing Jiaotong University, the M.Eng. degree from the Institute of Computing Technology, Chinese Academy of Science, and the Ph.D. degree from the University of Electro-Communications, Japan. Her main research interests include natural language processing, machine translation and machine learning. She has published more than 30 papers including over 10 papers in international journals such as a variety of IEEE and Natural Language Engineering. Prof. Zhang received the first prize of National Science and Technology Progress Award. She served on the editorial board of numerous journals and several conferences, such as Journal of Chinese Information Processing, the IEICE Transactions, MT Summit 2011, Coling 2014.