An Approach of Hybrid Hierarchical Structure for Word Similarity Computing by HowNet

Jiangming Liu
Beijing Jiaotong Universtity
jmliunlp@gmail.com

Jinan Xu
Beijing Jiaotong Universtity
jaxu@bjtu.edu.cn

Yujie Zhang
Beijing Jiaotong Universtity
yjzhang@bjtu.edu.cn

Abstract

Word similarity computing is an important and fundamental task 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. To overcome this problem, this paper proposes an approach of hybrid hierarchical structure computing Chinese word similarity to achieve fine-grained similarity results with HowNet 2008. The experimental results prove that the method has a better effect on computing similarity of synonyms and antonyms including nouns, verbs and adjectives. Besides, it performs stably on standard data provided by SemEval 2012.

1 Introduction

Word similarity computing plays an important role in various fields, such as Natural Language Understanding and Cognitive Science (Bunescu and Huang, 2010b; Mohler et al., 2011; Wang and Wan, 2011;). Moreover, it is a pivotal method in Word Sense Disambiguation (WSD).

Two main types of word similarity computing methods have been proposed. One is usually based on the thesaurus. The methods of this type utilize the structure of thesaurus (Liu and Li, 2002; Ge et al., 2010) with 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 methods are based on large-scale corpora with some inevitable disadvantages, such as the frequent need of large-scale corpora, noise, low search efficiency etc. (Nakov and Hearst, 2008). Therefore, it is fine to create a refined thesaurus with Internet resource or large-scale corpora (Morita et al., 2011; Navigli and Ponzetto, 2010; Davidov and Rappoport, 2010) as an interim for computing word similarity.

WordNet is deemed to be very valuable thesaurus. 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 lack of systematic resource. HowNet is also a valuable bilingual knowledge thesaurus organized by Zhongdong Dong.

HowNet uses a markup language called KDML to describe word's concept which facilitates computer processing (Li et al., 2012). 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 sememes are the most basic and the smallest units which cannot be easily divided (Liu and Li, 2002), and they are extracted from about six thousands of Chinese characters (Dong and Dong, 2006). An example of one DEF of *saleslady* can be described as a tree-like structure (Figure 1). The details of description in HowNet can be accessed in the paper (Dong, 2002).

In closely related works, Liu (2002) proposed an up-down algorithm on HowNet 2000 and achieved a good result. Li (2012) proposed an algorithm based on the hierarchic DEF description of words on HowNet 2002. In HowNet 2008, hierarchic DEF (Dong and Dong, 2002) definition is involved not only in words, but also in sememes. The algorithm proposed by Liu is useful, especially in example-based machine translation. The algorithm proposed by Li is detailedly experimented only in synonyms. The algorithm proposed in this paper fuses hierarchic DEF definition of sememe and hierarchic structure of sememe. It performs better and more stably both between the high similarity words namely synonyms and between the vague similarity words.

The remainder of the paper is organized as follows: Section 2 describes our algorithm in detail. Section 3 presents the experimental results and comparison. In the last section, conclusions are put forward and future work is discussed.

International Joint Conference on Natural Language Processing, pages 927–931,
Nagoya, Japan, 14-18 October 2013.

2 Similarity Computing

2.1 DEF similarity computing

The hierarchy of DEF is introduced as a tree-like structure. Due to different relation on the edge of trees, computing DEF similarity, unlike conventional tree similarity is one of our core works.

The 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 2.2.

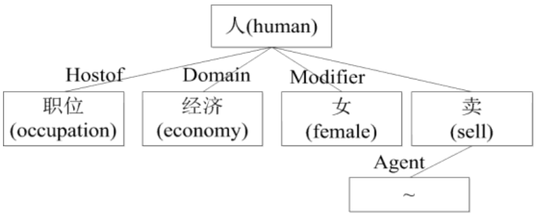

Figure 1. DEF hierarchy of *saleslady*

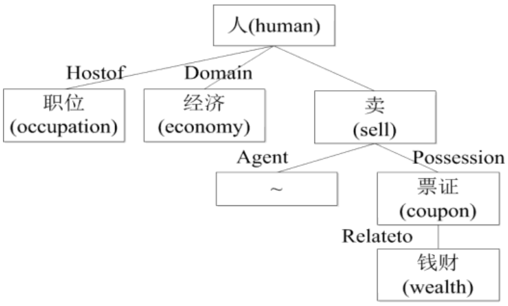

Figure 2. DEF hierarchy of *conductor*

For relation similarity, we take *saleslady* (Figure 1) and *conductor* (Figure 2) for example (Li, 2012) which are similar on morphology. When computing a pair of nodes similarity, such as root nodes, they 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 *δ*. Every pair of nodes should be calculated in DEF tree in the same layer as formula (1).

$$Sim_{node}(S_1,S_2)=\beta_{rela}\frac{1}{N}\sum_{i=1}^{N}Sim_{rela\_i}(S_1,S_2)+\beta_S Sim_s(S_1,S_2) \quad (1)$$

Where, N denotes N different kinds of relations, $Sim_{rela\_i}(S_1, S_2)$ 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 $\beta_{rela}>=0$, $\beta_s>=0$, $\beta_{rela}+\beta_s=1$. Bottom-to-up algorithm will be used to recursively compute DEF similarity in order to achieve the root node similarity as the DEF similarity.

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

The key point of DEF similarity computing method is not only taking the migration process of the nodes in the DEF tree into consideration (Li, 2012), but also using the relation between children nodes and their respective father node. In this way, the structure information from the DEF tree can be fully exploited.

However, there are special sememe (Attribute Sememe and Secondary Feature Sememe) whose weights are so high that the similarity unreasonably increases. Therefore, the formula (2) derived from the formula (1) is used to compute node similarity with a penalty factor ε.

$$Sim_{node}(S_1,S_2)=\beta_{rela}\frac{1}{N}\sum_{i=1}^{N}Sim_{rela\_i}(S_1,S_2)+\varepsilon\cdot\beta_S Sim_s(S_1,S_2) \quad (2)$$

2.2 Sememe similarity computing

The sememes are also described by DEF in HowNet2008. Therefore, sememe similarity (Sim_s (S_1, S_2)) can be divided into two parts, namely structure similarity and DEF similarity.

2.2.1 Structure similarity between sememes

In related works, many features of tree, such as the distance, depth and the least common nodes (LCN) in tree, have been used. This paper uses formula (3) below to compute structure similarity of sememe similarity

$$StructSim(S_1,S_2)=\frac{\alpha\cdot(depth(S_1)+depth(S_2))}{\alpha\cdot(depth(S_1)+depth(S_2))+dist(S_1,S_2)+|depth(S_1)-depth(S_2)|} \quad (3)$$

Where, $depth(S_1)$ represents depth of S_1 in sememe tree, and $dist(S_1,S_2)$ is the distance between S_1 and S_2 in sememe tree. It is clear that structure similarity of sememes increases with shorter distance between the sememes and a smaller difference in depth.

In the process of computing structure similarity, if there exists an 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 “-”. However, antonym relation and converse relation listed in HowNet document are too strict. Synonym Dictionary is used to extend antonym and converse relation.

2.2.2 DEF similarity between sememes

A special phenomenon exists in two aspects. On the one hand, in the process of computing DEF similarity, sememe similarity computing is needed. On the other hand, in the process of computing sememe similarity, DEF similarity computing is needed. This phenomenon brings about a cyclical calculation. In order to terminate infinite cyclical calculation, cyclical calculation will be processed only twice using formula (4)

$$Sim_s(S_1,S_2)=\begin{cases} StructSim(S_1,S_2) & if\ \ last\ \ circle \\ \beta_{struct} StructSim(S_1,S_2)+\beta_{DEF} Sim_{DEF}(S_1,S_2) & if\ \ not\ \ last\ \ circle \end{cases} \quad (4)$$

Where, *StructSim* (S_1, S_2) denotes structure similarity, and β_{struct}>=0, β_{DEF}>=0, $\beta_{struct}+\beta_{DEF}=1$. Sim_{DEF} (S_1, S_2) equals 1 if there is no DEF description of sememe in both S_1 and S_2. Convergence with cyclical calculation instead of twice will be researched in our future work.

2.3 Word similarity computing

Formula (5) below will be used to compute similarity between words containing one or more DEF description by

$$Sim_w(W_1,W_2)=\pm \max_{i=1...n,j=1...m} \left| Sim_{DEF}(S_{1i},S_{2j}) \right| \quad (5)$$

Where, S_{1i} is the *i*-th DEF of word W_1, S_{2j} is the *j*-th DEF of word W_2, "+" and "-" depend on the flag (section 2.2.1) of max DEF similarity. In formula (5), we choose maximum DEF similarity as word similarity by default.

3 Experiment and Comparison

General parameters in experiments derive from Liu's and Li's. The special parameters are optimized with greedy algorithm. Table 1 gives all the parameters of experiment.

3.1 Nouns and Verbs experiment

The result of our approach contrasted with Liu's and Li's is shown in Table 2. In Liu's approach, the similarity between words, such as pair of "man" and "father" and pair of "pink" and "crimson", is unreasonable. 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" with a flag "-" marking antonym.

general parameter	α	δ	β_{rela}	β_s
value	1.6	0.1	0.3	0.7
special parameter	β_{struct}	β_{DEF}	ε	
value	0.4	0.6	0.1	

Table 1. Parameters of experiment

3.2 Adjectives experiment

Li's algorithm and Liu's algorithm never take antonym relation into consideration. Jiang (2008) extends Liu's algorithm by using antonym relation. Table 3 shows that our result is much better than Jiang's result in many words. As we know, "beautiful" and "shifty-eyed" is strictly a pair of antonyms, and "shifty-eyed" is "ugly" but not vice versa.

Word 1	Word 2	Liu's result	Li's result	Our result
男人 (man)	女人 (woman)	0.8611	0.8955	-0.9957
男人 (man)	父亲 (father)	1.0000	0.8902	0.8904
男人 (man)	母亲 (mother)	0.8611	0.7857	-0.8875
粉红色 (pink)	深红色 (crimson)	1.0000	0.8500	-0.9829
名声 (repute-tion)	硬度 (hardness)	0.6176	/	0.2585
三伏 (hot)	冬眠 (hibernate)	0.0429	/	-0.6555

Table 2. Comparison of nouns and verbs

Word 1	Word 2	Jiang's result	Our result
美丽 (beautiful)	丑陋 (ugly)	-1.0000	-1.0000
美丽 (beautiful)	贼眉鼠眼 (shifty-eyed)	-1.0000	-0.9662
美丽 (beautiful)	优雅 (elegant)	0.7884	0.9264
舒服 (comfortably)	残疾 (handicap)	-0.0664	-0.7989
勇敢 (brave)	坚强 (strong)	0.7884	0.9500

Table 3. Comparison of adjectives

3.3 Synonyms experiment

In synonyms experiment, nearly 8000 pairs of words are randomly chosen as experimental data. The result (Figure 3) illustrates the effectiveness of our approach, since most of synonyms similarity is very high. Table 4 shows that our approach performs better than Li's in computing similarity of synonyms.

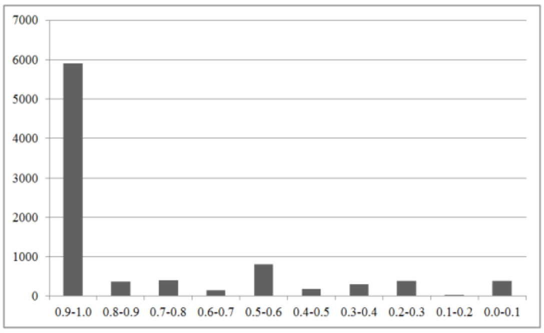

Figure 3. Result of synonyms experiment

	sim>0.9	sim>0.8	sim>0.7
Li's approach	60.89%	68.75%	72.85%
Our approach	66.05%	70.21%	74.76%

Table 4. Percentage of synonyms in different ranges

3.4 Antonyms experiment

Nearly 3000 pairs of antonyms are crawled from web resource for experiment. And the experimental results of antonyms come from two parts. One is the absolute value of antonyms experimental result denoting antonymous degree that is shown in Figure 4, and the other one is the flag "-" (section 2.2.1) marking antonym. Table 5 shows the percent of antonym in different ranges of similarity. Table 6 shows the number of pairs of antonyms with flag "-" by our approach.

The experimental results prove the high effectiveness of our approach of computing word similarity for most of antonyms similarity. However, it performs not very well in finding the flag "-" which marks antonym. With the development of HowNet, our approach will perform better.

Absolute similarity	>0.9	>0.8	>0.7
	50.02%	61.89%	68.70%

Table 5. Percentage of antonyms in different ranges

method	number in 3000 pairs
Original	863
Extend-Antoyms	966

Table 6. Number of antonyms with flag "-"

3.5 SemEval experiment

The datasets of Evaluating Chinese Word Similarity task In SemEval 2012 is used as the experimental data, of which the values are normalized as [0, 1]. The experimental data (130 pair words) covers similarity ranging from 0 to 1. Experimental data are sequenced by their similarity from high to low. The result of experiments is shown in Figure 5.

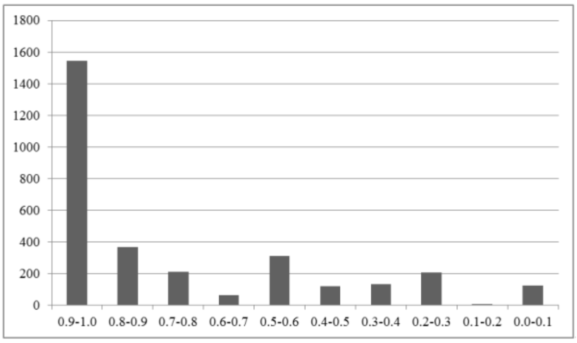

Figure 4. Result of antonyms experiment

Compared with Liu's method, the result shows that in the pairs of high similarity words, the difference of similarity is nearly 0.095. Besides, the largest difference is lower than 0.1. In Figure 5, the low difference value (0.01) between the highest difference and lowest difference is verified that the approach proposed by this paper is effective and stable in different range of similarity.

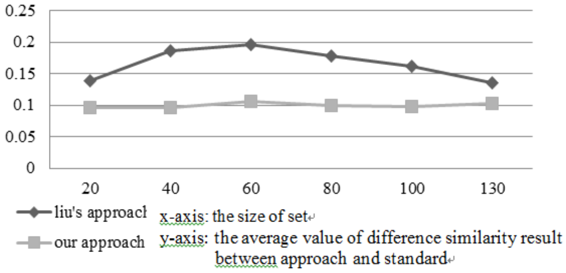

Figure 5. Comparison of experimental results

4 Conclusions and Future work

This paper proposes a new approach for computing word similarity between Chinese words using HowNet. The contribution can be concluded below. (1) Improve the accuracy of similarity by using EF description in sememe hierarchy; (2) substantiate that different kinds of sememe describe DEF in different weight; (3) use the Synonym Dictionary to alleviate strict limitations in antonym and converse relation.

Due to the importance of word context, in future, for documents, a method to choose suitable DEF for the word is necessary depending on context instead of maximum DEF similarity. Moreover, the alignment between sub-description of DEF is meaningful in computing semantic similarity. We will pay extra attention to sub-tree alignment. Based on these, we will optimize parameters for various applications.

References

Razvan Bunescu and Yunfeng Huang. (2010b). A utility driven approach to question ranking in social QA. In Proceedings *of The 23rd International Conference on Computational Linguistics (COLING 2010)*, 125–133.

Michael A.G. Mohler, Razvan Bunescu and Rada Mihalcea. (2011). Learning to Grade Short Answer Questions using Semantic Similarity Measures and Dependency GraphAlignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 752–762.

Lan Wang and Yuan Wan. (2011). Sentiment Classification of Documents Based on Latent Semantic Analysis. In *Communications in Computer and Information Science*, (176), 356-361

Qun Liu and Sujian Li, (2002) Word Semantic Similarity Computing Based on HowNet, *Computational Linguistics and Chinese Language Processing*, (7): 59-76.

Bin Ge, Fangfang Li, Silu Guo and Daquan Tang. (2010). The Research on Lexical Semantic Similarity Computing based on HowNet[J]. *Application Research of computers*, 27(9): 3329-3333

Preslav Nakov and Marti A. Hearst (2008). Solving relational similarity problems using theweb as a corpus. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, 452-460

Kazuhiro Morita, Shuto Arai, Hiroya Kitagawa, Masao Fuketa and Jun-ichi Aoe. (2011) Dynamic Construction of Hierarchical Thesaurus using Cooccurrence Information. *The 2nd International Conference on Networking and Information Technology IPCSIT*, Singapore

Roberto Navigli and Simone Paolo Ponzetto. (2010). BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Dmitry Davidov and Ari Rappoport (2010). Automated Translation of Semantic Relationships. In *Proceedings of The 23rd International Conference on Computational Linguistics (COLING 2010)*, 241-249.

Zhengdong Dong and Qiang Dong. (2002) Introduction to HowNet, http://www.keenage.com.

Hua Li, Changle Zhou, Min Jiang, Ke Cai. (2012). A hybrid approach for Chinese word similarity computing based on HowNet. *Automatic Control and Artificial Intelligence (ACAI 2012)*, 80-83

Peng Jin, Yunfang Wu. (2012). SemEval-2012 task 4: evaluating Chinese word similarity. In Proceeding of the First Joint Conference on Lexical and Computational Semantics. (1): 374-377

Min Jiang, Shibin Xiao, Hongwei Wang and Shuicai Shi. (2008). A improved Semantic Similarity Computing based on HowNet[J]. In Journal of Chinese information processing, 22(5): 84-89

Zhengdong Dong and Qiang Dong. (2006) HowNet and the Computation of Meaning, *World Scientific Publishing.*

Feng Li, Fang Li. (2007) An New Approach MeasuringSemantic Similarity in Hownet 2000, Journal of Chinese Information processing.