http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (20): 5685-5688, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.5685.5688

Dependency Syntactic Tree Supported Sentence Similarity Computing

[1]Xiong Jing, [1]Liu Yun-Tong and [2]Yuan Dong
[1]School of Computer and Information Engineering, Anyang Normal University, Anyang, 455000, Henan, China
[2]High-Performance Server and Storage Technologies, State Key Laboratory, Jinan, 250100, Shandong, China

Abstract: In most fields of Natural Language Processing (NLP), sentence similarity computing plays an important role. At the same time, the syntactic similarity computing is the basis of sentence similarity computing. This study introduces a dependency syntactic tree similarity computation method based on multi-features. The method studies many features of dependency syntactic tree including word and word's POS of each node and the dependency type between them. Then the similarity algorithm is proposed after comprehensively analyzing all the features. The experimental result is satisfied as this method describes dependency syntactic tree more comprehensively and accurately.

Key words: Similarity computing, syntactic dependency, multi-features, NLP, sentence processing

INTRODUCTION

Similarity is a complex concept and it is wildly discussed in semantics, philosophy and information science, etc. Now sentence similarity computing plays an increasingly important role in text-related research and applications in areas such as text mining (Li *et al*., 2006). It is the basis of many applications, such as snippet extraction, image retrieval, question-answer model and document retrieval (Gu *et al*., 2012). Syntactic similarity is the base of sentence similarity and the dependency-based representations in natural language parsing are increasing in recent years. The fundamental notion of dependency is based on the idea that the syntactic structure of a sentence consists of binary asymmetrical relations between the words of the sentence (Nivre, 2005).

There are some kinds of dependencies: Semantic dependencies (Melcuk, 2003), morphological dependencies (Ahonen *et al*., 1997), prosodic dependencies (GroB, 2011) and syntactic dependencies (Gibson, 1998). Dependency syntactic tree is made use of to find similar questions within the predetermined categories (Lian *et al*., 2013). There are few researches studying on syntactic similarity computation and most of them focus on words (Sagae and Gordon, 2009), not the whole structure.

In this study, we introduce a new method to compute the similarity between two dependency syntactic trees which considering not only word but also word's POS, dependency type and correspondence between the two trees. This method can be used to measure the similarity between two dependency syntactic trees and to estimate dependency syntactic parsers by comparing the similarity between the output and manually annotated result.

DEPENDENCY SYNTACTIC

Dependency grammar is proposed by French linguist L.Tesniere in his book Eléments de la syntaxe structurale in 1959 which had a profound impact on Linguistics and is especially respected in Computational Linguistics. Dependency grammar reveals the syntactic structure by analyzing the dependency relationship among compositions of a sentence. In dependency grammar theory, verb is considered as the center or root of a sentence which dominate the other parts but verb itself is not dominated by any composition (Green and Dorr, 2004) and all dependency relationship can be described by one kind of dependency type.

The structure of a sentence can be represented as two forms in dependency grammar theory, one is syntactic tree and the other is relationship collection. As an example, the sentence "I use spring framework in my webapplication." can be described as a tree showed in Fig. 1.

The relationship collection of sentence "I use spring framework in my webapplication." is shown Table 1.

Corresponding Author: Xiong Jing, School of Computer and Information Engineering,Anyang Normal University, Anyang, 455000, Henan, China

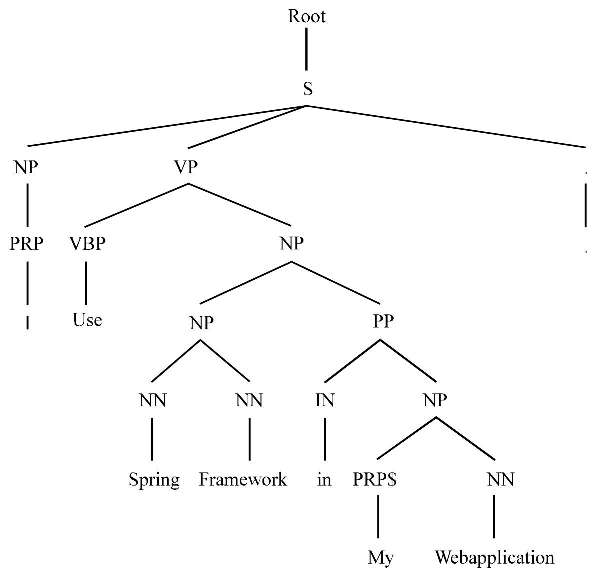

Fig.1: A syntactic tree sample

Table 1: Relationship collection of Fig. 1

nsubj(use-2, I-1)
root(ROOT-0, use-2)
nn(framework-4, spring-3)
dobj(use-2, framework-4)
poss(webapplication-7, my-6)
prep_in(framework-4, webapplication-7)

Our method uses the relationship collection form to compute the similarity between two syntactic trees.

SIMILARITY COMPUTATION OF DEPENDENCY RELATIONSHIP

The similarity of two relationship collections is based on the similarity of two relationships. We use P(H/H-POS,C/C-POS,D) to present one relationship. What it indicates is that word C is dependent on word H, H-POS is the POS of H, C-POS is the POS of C and D is the dependency type between C and H. The five features in P have different importance.

By the dependency grammar theory, one word can only depend on a specific word but it can have many words depended on it. So in P, C is more important than H. On the other hand, one word has few kinds of POS but one POS includes lots of words. So 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. Finally we get the order of weight for the five features: C>H>D>C-POS>H-POS.

Suppose there are two relationships 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), consider their five corresponding features, 1 for equal while 0 for not equal. Then order these five numbers by their weights, we can get a binary number $(bbbbb)_2$ which is ranged from 0 to 31 and 0 means the two relationships are exactly the same while 31 means the two are completely different. Based on this binary number, we define the similarity between P_1 and P_2 as:

$$SR(P_1, P_2) = \frac{(bbbbb)_2}{(11111)_2} \quad (1)$$

Assumed within P_1 and P_2, there are $C_1 = C_2$, C_1-POS ≠ C_2-POS, $H_1 \neq H_2$, H_1-POS ≠ H_2-POS, $D_1 = D_2$, then the binary number is $(10100)_2$ and the similarity between P_1 and P_2 is:

$$SR(P_1, P_2) = \frac{(10100)_2}{(11111)_2} = \frac{20}{31} = 0.6452$$

Let dependency relationship collections A = (a_1, a_2,..., a_n) and B = (b_1, b_2, ..., b_n). Since, the symmetry of similarity computation, we may assume that the size of A is smaller or equal than B that is n = m.

For the similarity computation between A and B, we must decide how the relationships from A map the relationships from B. For each $a_i \neq A$, $1 = i = n$, we can find $b_j \neq B$, $1 = j = m$. We assumed that different b_j for different a_i and then there are n!/(m-n)! kinds of ways how collection A maps collection B.

Then we discuss a certain kind of way marked as Ω_k, $1 = k = n!/(m-n)!$. In Ω_k, for a certain relationship a_i, there is a relationship b_j which we can mark as $b_j = \Omega_k(a_i)$. Then we define the similarity of Ω_k:

$$Sim(\Omega_k) = \frac{\sum_{i=1}^{n}[SR(a_i)]\Omega_k(a_i)}{m} \quad (3)$$

Based on $Sim(\Omega_k)$, we define the similarity of two relationship collections A and B as:

$$SRC(A, B) = Max(Sim(\Omega_k)) \quad (4)$$

where, the value of k is.

We choose the way which makes the similarity between A and B gets the maximum. The similarity computed in that way is considered as the similarity between A and B.

For example, sentence S_1 "I use spring framework in my webapplication." whose syntactic structure showed in Table 1 and sentence S_2 "I create my webapplication based on EJB." whose syntactic tree is shown in Fig. 2.

Table 2 shows the dependency relationship collections similarity computing between sentence S_1 and S_2.

Table 2: Dependency relationship similarity computing of S1 and S2

Relationship collection of S_1	Relationship collection of S_2	SR_{max}(S1,S2) (%)
a1:nsubj(use/VBP, I/PRP,nsubj)	b1:nsubj(create/VBP, I/PRP, nsubj)	a1|b1 = 48.39
a2:root(ROOT/-, use/VBP,root)	b2:root(ROOT/-, create/VBP, root)	a2|b2 = 74.19
a3:nn(framework/NN, spring/NN,nn)	b3:poss(webapplication/NN,my/PRP$,poss)	a3|b3 = 6.45
a4:dobj(use/VBP, framework/NN,dobj)	b4:dobj(create/VBP,webapplication/NN,dobj)	a4|b4 = 22.58
a5:poss(webapplication/NN,my/PRP$,poss)	b5:prepc_based_on(create/VBP, on/IN,prep)	a5|b3 = 100
a6:prep_in(framework/NN, webapplication/NN,prep)	b6:pobj(create/VBP, EJB/NNP,pobj)	a6|b4 = 29.03

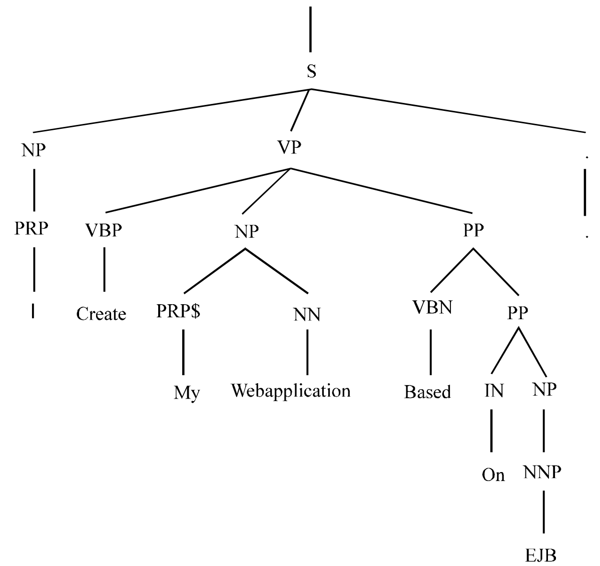

Fig. 2: Syntactic tree of sentence S_2

Then based on Eq. 2-4, we can get the similarity between these two syntactic structures:

$$SRC(S1.S2) = \frac{0.4839 + 0.7419 + 0.0645 + 0.2258 + 1 + 0.2903}{6} = 46.77\% \quad (5)$$

EXPERIMENTS AND ANALYSIS

Experiment 1: We choose 40 sentences which will be calculated to compare similarity with a specified sentence. There is one standard sentence and the others which are similar with the standard sentence in syntactic structure are manually ordered by the similarity. Then we order the sentences with our similarity computation algorithm. The top 10 most similar sentences are showed in Table 3.

The results show that our algorithm has almost the same order with the manual one in syntactic structure. Our analysis of the results shows that when the syntactic structure of test sentence is consistent with the standard one and the test sentence has lots of same words with the standard sentence simultaneously, their similarity is high. We analyzed the sentence syntactic tree and discovered that when the root node which is the central word of sentence in dependency grammar is incorrectly recognized or not the same, the similarity between the two structures is low or even zero. This algorithm could not handle polysemous word well such as "spring" in standard sentence and No. 6 sentence.

Table 3: Top 10 most similar sentences

No.	Similar sentences	Similarity (%)
1	I create my webapplication based on EJB.	46.77
2	Officials use loud hailers to call for calm.	33.64
3	The sports in my school are great.	26.34
4	I made a few phone calls.	23.11
5	Lizzie bought herself a mountain bike.	22.04
6	Trees put forth buds in spring.	20.97
7	You should check the order of the middleware.	18.43
8	Researches of sentences similarity computation method based on Hownet.	17.97
9	Those rockets landed in the desert.	17.74
10	Web development has been a growing industry.	17.51

Table 4: Similarity algorithm precision, recall and F value

Method	Precision (%)	Recall (%)	F1 (%)
VSM	46.32	76.12	57.59
Ours	57.65	81.29	67.46

Experiment 2: Given another 25 sentences, they are divided into 5 groups: computer, literature, philosophy, internet and industry. Each group includes 5 similar sentences. Put these 25 sentences into a test set including 200 noise sentences. To take each one of these 25 sentences as a standard sentence and compute the similarity with others. After order the sentence similarity, choosing the top 4 most similar sentences. If the checked sentences belong to the group of the standard sentence, it means they are right sentences. Comparing with the SVM, the precision, recall and F value are shown in Table 4.

Experimental results show that our approach compared VSM method have greatly improved the precision and recall. The main factors affecting the algorithm performance are sentence length, dependency parsing accuracy and center word differences. The more consistent of compared sentences in syntactic structure and the more common words they contain, the higher the similarity. And in this condition, the smaller the difference of dependencies numbers, the higher the similarity.

CONCLUSION

This study introduces a dependency syntactic tree similarity computation method based on multi-features.

This method starts from dependency relationship, matches relationships from two collections to get the maximum of the similarity and then finds the average of all similarity of relationships as the similarity of the two dependency syntactic trees. In this method, we consider five features of dependency syntactic tree including word and its POS of each node and the dependency type between them and then comprehensively analyze the similarity relationship of all the syntactic structures. However, this algorithm does not handle the semantic link within dependencies. Semantic-based dependency analysis is our future main research content.

ACKNOWLEDGMENTS

This research is supported by Development projects of Henan province science and technology (132102210264) and Henan Province Education Office Humanities and Social Science Project (2013-GH-383).

REFERENCES

Ahonen, H., O. Heinonen, M. Klemettinen and A.I. Verkamo, 1997. Applying data mining techniques in text analysis. Report C-1997-23, Departement of Computer Science, University of Helsinki.

Gibson, E., 1998. Linguistic complexity: Locality of syntactic dependencies. Cognition, 68: 1-76.

Green, R. and B. Dorr, 2004. Inducing a semantic frame lexicon from WordNet data. Proceedings of the 2nd Workshop on Text Meaning and Interpretation, July 21-26, 2004, Association for Computational Linguistics Stroudsburg, Barcelona, pp: 65-72.

GroB, T., 2011. Clitics in dependency morphology. Proceedings of the International Conference on Dependency Linguistics Depling 2011, September 5-7, 2011, Barcelona, pp: 58-68.

Gu, Y., Z. Yang, M. Nakano and M. Kitsuregawa, 2012. Towards efficient similar sentences extraction. Proceedings of the 13th International Conference on Intelligent Data Engineering and Automated Learning-Ideal, Natal, Brazil, August 29-31, 2012, Springer, Berlin, Heidelberg, pp: 270-277.

Li, Y., D. Mclean, Z.A. Bandar, J.D. O'Shea and K. Crockett, 2006. Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng., 18: 1138-1150.

Lian, X., X. Yuan, X. Hu and H. Zhang, 2013. Finding similar questions with categorization information and dependency syntactic tree. Proceedings of the 14th International Conference on Web-Age Information Management, June 14-16, 2013, Springer, Berlin, Heidelberg, pp: 607-612.

Melcuk, I., 2003. Levels of Dependency in Linguistic Description: Concepts and Problems. In: Dependency and Valency: An International Handbook of Contemporary Research, Agel, V., L. Eichinnger, H.W. Eroms, P. Hellwig, H.J. Herringer and H. Lobin (Eds.). Vol. 1, W. de Gruyter Publisher, Berlin, New York, pp: 188.

Nivre, J., 2005. Dependency grammar and dependency parsing. MSI Report No. 5133. pp: 1-32.

Sagae, K. and A.S. Gordon, 2009. Clustering words by syntactic similarity improves dependency parsing of predicate-argument structures. Proceedings of the 11th International Conference on Parsing Technologies, 7 October 2009, Association for Computational Linguistics, USA., pp: 192-201.