International Journal of Hybrid Information Technology
Vol.8, No.10 (2015), pp.205-216
http://dx.doi.org/10.14257/ijhit.2015.8.10.19

An Optimization for Hybrid Semantic Similarity Computation

Zhixiao Wang[1,2], Xiaofang Ding[2] and Ying Huang[2]

[1]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China*
[2]*College of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China*
softstone416@163.com

Abstract

Semantic similarity computation is of great importance in many applications such as natural language processing, knowledge acquisition and information retrieval. In recent years, many concept similarity measures have been developed for ontology and lexical taxonomy. Generally speaking, ontology concepts semantic similarity computation is tedious and time-consuming. This paper puts forward an optimization algorithm to simplify semantic similarity computation. The optimization algorithm utilizes hierarchical relationship between concepts to simplify similarity computation process. Simulation experiments showed the optimization algorithm could make similarity computation simple and convenient, and similarity computation speed was improved by one time. The more complexity an ontology structure, and the bigger the maximum depth of ontology, the more significantly the performance improved.

Keywords: *Ontology, Semantic Similarity, Concepts Hierarchical Relationship, Optimization Algorithm*

1. Introduction

Semantic similarity is a generic issue in the variety of application areas of Artificial Intelligence and Natural Language Processing. Semantic similarity can be exploited to improve Information Retrieval performance [1], to carry out query expansion [2], to perform word-sense disambiguation [3]. Categorization or clustering [4] algorithms also rely on semantic similarity measures to detect and group similar subjects. Semantic similarity is also useful for computational biology and bioinformatics [5].

Ontology and lexical taxonomy provide a formal specification of a shared conceptualization, which have been of great interest for the semantic similarity research community as they offer a structured and unambiguous representation of knowledge in the form of conceptualizations interconnected by means of semantic pointers. These structures can be exploited in order to assess the degree of semantic proximity between terms. In recent years, ontology has been extensively exploited to compute semantic similarity, and a number of methods have been developed in literature. These methods can be classified on the basis of information source they exploit [6].

Edge counting approach. This approach takes ontology as directed graph, and semantic similarity can be assessed by counting the number of edges in the graph path between two concepts. Edge counting approach is intuitive, However, several limitations hamper its performance [7]. This approach only consider the shortest path between concept pairs, other features also influencing the concept semantics, such as the number and distribution of common and non-common taxonomical ancestors are neither concepts, many of the taxonomical knowledge explicitly modelled in the ontology is omitted.

Information Content approach. This approach associates appearance probabilities to each concept in the taxonomy. Semantic similarity depends on the amount of shared

ISSN: 1738-9968 IJHIT

information between two terms, which represented by their Least Common Ancestor (LCA) in an ontology. The limitation of this approach due to its dependency on corpora [8], they require big and fine grained ontology with a detailed taxonomical structure in order to properly differentiate concept's IC.

Hybrid approach. This approach combines multiple information sources to calculate semantic similarity, which can make the similarity results more accurate.

Li, Bandar, and Mclean [9] proposed another hybrid approach. The factors of path length, depth and density are considered in the assessment, which can be mathematically expressed as:

$$sim(c_i, c_j) = \begin{cases} e^{-\alpha l} \cdot \dfrac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & c_i \neq c_j \\ 1 & otherwise \end{cases} \quad (1)$$

In formula (1), l is the shortest path length between concept c_i and concept c_j, h is the depth of the lowerest common ancestor in the ontology. α and β are parameters scaling the contribution of shortest path length and depth, respectively. The optimal values are as follows: α =0.2, β =0.6.

David Sanchez, *et al.*, [7] found that the method of Li, *et al.*, significantly outperforms traditional similarity measures. Angelos Hliaoutakis, *et al.*, [10] also pointed out this method is particularly effective in semantic similarity computation. The method of Li, *et al.*, has been used in many applications. For example, Wei Song, *et al.*, [8] proposed a text clustering algorithm based on this semantic similarity measure. It can be seen that the method of Li, *et al*., is a very successful hybrid approach.

For the method of Li, *et al.*, each concepts pair similarity computation needs two parameters: the shortest path length of concepts and the depth of the lowerest common ancestor in the ontology. Obtaining these values is time-consuming. Therefore, it is quite necessary to reduce the computational complexity of semantic similarity computation.

This paper puts forward an optimization algorithm for the semantic similarity models of Li, *et al.*, The optimization algorithm utilizes hierarchical relationship between concepts to simplify similarity computation process. Based on the semantic similarity of one concept pair, the optimization algorithm can give semantic similarity of arbitrary concept pair in the ontology. Simulation experiments showed that the computation complexity was reduced considerably, and similarity computation speed was improved by one time.

This paper is organized as follow: section 2 is related works; section 3 focuses on the feasibility analysis of similarity computation optimization, the optimization algorithm description and complexity analysis; section 4 is simulation experiment and results; and section 5 comes the conclusion of this paper.

2. Related Works

David Sanchez, *et al.*, [7] surveyed and compared most of the ontology-based similarity measures developed in recent years.

Rada, *et al.*, [11] provided an edge counting approach.

$$sim(c_1, c_2) = 2 \times Max - Dis(c_1, c_2) \quad (2)$$

where, $sim(c_1, c_2)$ is the semantic similarity between concept c_1 and concept c_2; Max is the maximum depth of the taxonomy; $Dis(c_1, c_2)$ is the minimum number of edges separating c_1 and c_2.

Leacock and Chodorow [12] considered that the number of edges on the shortest path between two concepts should be normalized by the depth of a taxonomic structure, which is expressed:

$$sim(c_1, c_2) = -\log(Dis(c_1, c_2) / 2 \cdot Max) \tag{3}$$

Wu and Palmer [13] provided another edge counting approach. The measure mentioned the node that subsumes two concepts when computing the similarity between the two concepts, which can be expressed mathematically as follows:

$$sim(c_1, c_2) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3} \tag{4}$$

where, c_3 is the lowerest common ancestor of c_1 and c_2, N_1 is the minimum number of edges from c_1 to c_3, N_2 is the minimum number of edges from c_2 to c_3, N_3 is the depth of c_3.

Resnik [14] put forward an Information Content approach whereby the information shared by two concepts can be indicated by the concept that subsumes the two concepts in a taxonomy. The similarity between the two concepts c_1 and c_2 can be mathematically expressed as follows:

$$sim_{res}(c_1, c_2) = IC(LCA(c_1, c_2)) \tag{5}$$

where, $LCA(c_1, c_2)$ is the Least Common Ancestor of c_1 and c_2 in an ontology.

Lin's [15] semantic similarity model is the extension of Resnik's model, which measures the similarity between two nodes as the ratio between the amount of commonly shared information of the two nodes and the amount of information of the two nodes, which can be mathematically expressed as follows:

$$sim_{lin} = \frac{2 \times sim_{res}(c_1, c_2)}{IC(c_1) + IC(c_2)} \tag{6}$$

Pirro [16] proposed another model extended from Resnik's model. The similarity between two concepts is a function of common features between the two concepts minus those in each concept but not in another concept. By integrating Resnik's model, the similarity model can be mathematically expressed as follows:

$$sim(c_1, c_2) = \begin{cases} 3 \times sim_{res}(c_1, c_2) - IC(c_1) - IC(c_2) & if\ (c_1 \neq c_2) \\ 1 & if\ (c_1 = c_2) \end{cases} \tag{7}$$

Jiang and Conath [17] developed a hybrid model that uses the information-based theory to enhance the edge-based model.

$$Dis(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \times sim_{res}(c_1, c_2) \tag{8}$$

$$sim_{Jiang\&Conath}(c_1, c_2) = 1 - Dis(c_1, c_2) \tag{9}$$

Li, *et al.*, [9] proposed a hybrid semantic similarity model combining structural semantic information in a nonlinear model. The factors of path length, depth and density are considered in the assessment, as shown in formula (1).

Al-Mubaid and Nguyen [18] proposed another cluster-based hybrid model, which measures the common specificity of two terms by subtracting the depth of their LCS from the depth D_c of the cluster.

$$sim_{res}(c_1, c_2) = D_c - depth(LCS(c_1, c_2)) \tag{10}$$

As introduced above, Hybrid approach combines multiple information sources to calculate semantic similarity, such as shortest path length between compared words, information content, depth in the taxonomy hierarchy, and semantic density of compared words, *et al.*, which can make the similarity results more accurate.

The method of Li, *et al.*, is a very successful hybrid approach. This paper puts forward an optimization algorithm for this method. The optimization algorithm utilizes hierarchical relationship between concepts to simplify similarity computation process. Based on the semantic similarity of one concept pair, the optimization algorithm can give semantic similarity of arbitrary concept pair in the ontology.

3. Methods

3.1 Feasibility Analysis of Similarity Computation Optimization

Figure 1 shows parts of ontology concepts. Suppose that the semantic similarity $sim(X,B)$ between X and B, the shortest path length l between X and B and the depth h of the lowerest common ancestor are all known. Can we get the semantic similarity between X and sub-concept of B (for example, C) or the semantic similarity between X and parent concept of B (for example, A) ?

This paper investigates the problem in two cases: (1) X is not the sub-concept of B; (2) X is the sub-concept of B.

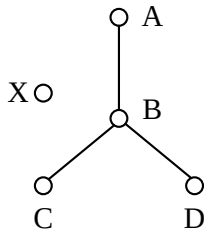

Figure 1. Parts of Concepts

3.1.1 X is not the Sub-concept of B : In this case, X may be the parent concept of B (as Figure 2(b) shows) or may not be the parent concept of B (as Figure 2(a) shows).

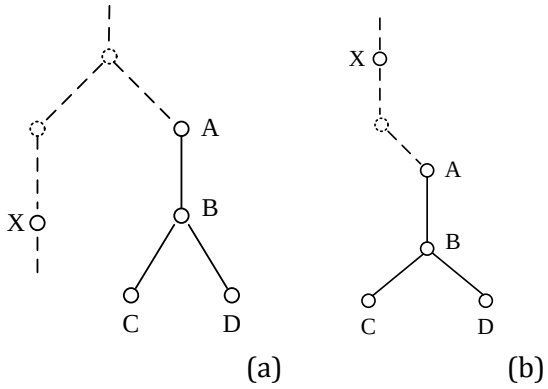

Figure 2. X is not the Sub-concept of B

(1) Semantic similarity $sim(X,A)$ between X and A.

In Figure 2, A is direct parent of B, the shortest path length between X and A is 1 less than that of between X and B, that is $l-1$. The lowerest common ancestor of X and A is the same as that of X and B, and so the depth of the lowerest common ancestor is h still.

Therefore:

$$\frac{sim(X,A)}{sim(X,B)} = \frac{e^{-\alpha(l-1)} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}}{e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}} = e^{\alpha} \tag{10}$$

That is:

$$sim(X,A) = e^{a} \cdot sim(X,B) \tag{11}$$

where, α is constant.

Formula (11) shows that, based on the product of $sim(X,B)$ and e^{α}, $sim(X,A)$ can be directly obtained.

(2) Semantic similarity $sim(X,C)$ between X and C.

In Figure 2, C is direct sub-concept of B, the shortest path length between X and C is 1 more than that of between X and B, that is $l+1$. The lowerest common ancestor of X and C is the same as that of X and B, and so the depth of the lowerest common ancestor is h still.

Therefore:

$$\frac{sim(X,C)}{sim(X,B)} = \frac{e^{-\alpha(l+1)} \cdot \dfrac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}}{e^{-\alpha l} \cdot \dfrac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}} = e^{-\alpha} \tag{12}$$

That is:

$$sim(X,C) = e^{-a} \cdot sim(X,B) \tag{13}$$

where, α is constant.

Formula (13) shows that we can directly obtain $sim(X,C)$ based on the product of $sim(X,B)$ and $e^{-\alpha}$.

3.1.2 X is the Sub-concept of B: (1) Semantic similarity $sim(X,A)$ between X and A.

X is sub-concept of B, and B is sub-concept of A, thus, X must be sub-concept of A. In Figure 3, the shortest path length between X and A is 1 more than that of between X and B, that is $l+1$. The lowerest common ancestor of X and B is B, the lowerest common ancestor of X and A is A. The depth of B is h, so, the depth of A is $h-1$.

Therefore:

$$\frac{sim(X,A)}{sim(X,B)} = \frac{e^{-\alpha(l+1)} \cdot \dfrac{e^{\beta(h-1)} - e^{-\beta(h-1)}}{e^{\beta(h-1)} + e^{-\beta(h-1)}}}{e^{-\alpha l} \cdot \dfrac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}}$$

$$= e^{-\alpha} \cdot (1 - \frac{2}{e^{2\beta(h-1)} + 1}) \cdot (1 + \frac{2}{e^{2\beta h} - 1}) \tag{14}$$

That is:

$$sim(X,A) = e^{-\alpha} \cdot (1 - \frac{2}{e^{2\beta(h-1)} + 1}) \cdot (1 + \frac{2}{e^{2\beta h} - 1}) \cdot sim(X,B) \tag{15}$$

where, α, β is constant, h is known. We can directly obtain $sim(X,C)$ by formula (15).

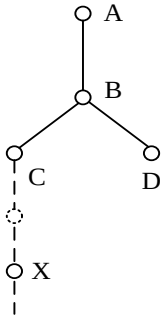

Figure 3. X is the Sub-concept of B

(2) Semantic similarity between X and the direct sub-concept of B.

1) X and the direct sub-concept of B locate same branch, such as C in Figure 3.

C is direct sub-concept of B, the shortest path length between X and C is 1 less than that of between X and B, that is $l-1$. The lowerest common ancestor of X and B is B, the lowerest common ancestor of X and C is C. The depth of B is h, so, the depth of A is $h+1$.

Therefore:

$$\frac{sim(X,C)}{sim(X,B)} = \frac{e^{-\alpha(l-1)} \cdot \frac{e^{\beta(h+1)} - e^{-\beta(h+1)}}{e^{\beta(h+1)} + e^{-\beta(h+1)}}}{e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}}$$

$$= e^{\alpha} \cdot (1 - \frac{2}{e^{2\beta(h+1)} + 1}) \cdot (1 + \frac{2}{e^{2\beta h} - 1}) \quad (16)$$

That is:

$$sim(X,C) = e^{\alpha} \cdot (1 - \frac{2}{e^{2\beta(h+1)} + 1}) \cdot (1 + \frac{2}{e^{2\beta h} - 1}) \cdot sim(X,B) \quad (17)$$

where, α, β is constant, h is known. We can directly obtain $sim(X,C)$ by formula (17).

2) X and the direct sub-concept of B locate different branch, such as D in Figure 3.

D is direct sub-concept of B, the shortest path length between X and D is 1 more than that of between X and B, that is $l+1$. The lowerest common ancestor of X and D is the same as that of X and B, and so the depth of the lowerest common ancestor is h still.

Therefore:

$$\frac{sim(X,D)}{sim(X,B)} = \frac{e^{-\alpha(l+1)} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}}{e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}} = e^{-\alpha} \quad (18)$$

That is:

$$sim(X,D) = e^{-a} \cdot sim(X,B) \quad (19)$$

where, α is constant.

Formula (19) shows that we can directly obtain $sim(X,D)$ based on the product of $sim(X,B)$ and $e^{-\alpha}$.

From above analyses, we can see that with the semantic similarity of one concept pair (such as $sim(X,B)$), by utilizing hierarchical relationship among ontology concepts, we can get semantic similarity of arbitrary concept pair in the ontology, the similarity computation process is greatly simplified.

3.2 Algorithm Description and Complexity Analysis

Input: ontology O, concept X, concept B, $X, B \in O$, but $X \neq B$, h, $sim(X,B)$;

Output: semantic similarity between X and all the other concepts in ontology O.

Algorithm Description:

SemanticSimilarity(X, B, $sim(X,B)$, h)

{

IF ($X \notin \sup class(B)$) // $\sup class(B)$ are sub-concept of B

$sim(X, direct\_\sup erclass(B)) = e^{a} \cdot sim(X,B)$

$sim(X, direct\_subclass(B)) = e^{-a} \cdot sim(X,B)$

ELSE

$sim(X, direct\_\sup erclass(B)) =$

$$e^{-\alpha} \cdot (1 - \frac{2}{e^{2\beta(h-1)} + 1}) \cdot (1 + \frac{2}{e^{2\beta h} - 1}) \cdot sim(X,B)$$

For each $direct\_subclass(B)$

IF ($direct\_subclass(B) \in \sup erclass(X)$)

$sim(X, direct\_subclass(B)) =$

$$e^{\alpha} \cdot (1 - \frac{2}{e^{2\beta(h+1)} + 1}) \cdot (1 + \frac{2}{e^{2\beta h} - 1}) \cdot sim(X, B)$$

ELSE

$sim(X, direct\_subclass(B)) = e^{-a} \cdot sim(X, B)$

End IF

End For

End IF

IF $direct\_\sup class(B)) \neq$ root concept

//does not arriving root concept

SemanticSimilarity(

$X, direct\_\sup erclass(B), sim(X, direct\_\sup erclass(B)), h-1)$

End IF

For each $direct\_subclass(B)$

IF $direct\_subclass(B) \notin$ leaf concepts

//does not arriving leaf concepts

SemanticSimilarity(

$X, direct\_subclass(B), sim(X, direct\_subclass(B)), h+1)$

End IF

End For

}

Complexity Analysis

For formula (1), the key steps of similarity computation are the shortest path length counting and the depth of the lowerest common ancestor judgment. Therefore, We define these two procedures as basic operation of similarity computation.

(1) Complexity analysis of original algorithm

For original algorithm, we must obtain l and h in every concept pair similarity computation. Therefore, the complexity of single concept pair is $O(2)$. Suppose ontology O has n concepts, the similarity matrix of these n concepts $S_{n\times n}$ is:

$$S_{n\times n} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1j} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2j} & \cdots & s_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ s_{i1} & s_{i2} & \cdots & s_{ij} & \cdots & s_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ s_{n1} & s_{n2} & \cdots & s_{nj} & \cdots & s_{nn} \end{bmatrix} \quad (20)$$

s_{ij} （$1 \le i \le n, 1 \le j \le n$） is semantic similarity between concept i and concept j. The semantic similarity matrix is symmetric matrix., that is $s_{ij} = s_{ji}$; The semantic similarity between one concept and itself is 1, that is s_{ii}=1. Thus, we only need to compute upper triangular matrix or lower triangular matrix of $S_{n\times n}$. The complexity is:

$$O(2 \cdot 1 + 2 \cdot 2 + \cdots + 2 \cdot (n-1)) = O(\frac{2n(n-1)}{2}) = O(n^2 - n) \quad (21)$$

(2) Complexity analysis of optimization algorithm

Optimization algorithm utilizes hierarchical relationship between concepts to simplify similarity computation process, which obtains l and h in one basic operation. Therefore, the complexity of single concept pair is $O(1)$, and all semantic similarity computation complexity is:

$$O(1\cdot 1+1\cdot 2+\cdots+1\cdot(n-1))=O(\frac{n^2-n}{2}) \tag{22}$$

3.3 Further Discussions

Semantic similarity may be influenced by relation types between concepts. For example, there are two directly connected concepts A and B. if the relationship between them is "equivalentClass", the semantic similarity is 1. If the relationship is "part-of", the semantic similarity is different.

In order to compute similarity more accurately, different relationship can be set different weight. The optimization idea of section 3 can also be applied to the weighted similarity computation. Here, we take Figure 2 as an example to analysis.

(1) Semantic similarity $sim(X,A)$ between X and A.

In Figure 2, A is direct parent of B, the shortest path length between X and A is less $\omega_{AB}\cdot 1$ than that of between X and B, that is $l-\omega_{AB}$, ω_{AB} is the relationship weight. The lowerest common ancestor of X and A is the same as that of X and B, and so the depth of the lowerest common ancestor is h still.

Therefore:

$$\frac{sim(X,A)}{sim(X,B)}=\frac{e^{-\alpha(l-\omega_{AB})}\cdot\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}}}{e^{-\alpha l}\cdot\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}}}=e^{\alpha\omega_{AB}} \tag{23}$$

That is:

$$sim(X,A)=e^{a\omega_{AB}}\cdot sim(X,B) \tag{24}$$

where, α and ω_{AB} is constant.

Formula (23) shows that we can directly obtain $sim(X,A)$ based on the product of $sim(X,B)$ and $e^{\alpha\omega_{AB}}$.

(2) Semantic similarity $sim(X,C)$ between X and C.

In Figure 2, C is direct sub-concept of B, the shortest path length between X and C is more $\omega_{BC}\cdot 1$ than that of between X and B, that is $l+\omega_{BC}$, ω_{BC} is the relationship weight. The lowerest common ancestor of X and C is the same as that of X and B, and so the depth of the lowerest common ancestor is h still.

Therefore:

$$\frac{sim(X,C)}{sim(X,B)}=\frac{e^{-\alpha(l+\omega_{BC})}\cdot\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}}}{e^{-\alpha l}\cdot\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}}}=e^{-\alpha\omega_{BC}} \tag{25}$$

That is:

$$sim(X,C)=e^{-a\omega_{BC}}\cdot sim(X,B) \tag{26}$$

where, α and ω_{BC} is constant.

Formula (25) shows that we can directly obtain $sim(X,C)$ based on the product of $sim(X,B)$ and $e^{-\alpha\omega_{BC}}$.

4. Results

In this section, we empirically compared our optimization algorithm with the original algorithm proposed by Li et al. Simulation program implemented with java and relative tools in the Linux environment. In order to furthest weak the influence of computer system running state and obtain relatively accurate results, all semantic similarity computation was carried out 1000 times. We define an evaluation criterion named optimization performance as follows:

$$\text{optimization performance} = \frac{T_1 - T_2}{T_1} \tag{27}$$

where, T_1 is average computation time of original algorithm, T_2 is our algorithm.

Firstly, we compute semantic similarity of ACMCSS ontology concepts (http://www.acm.org/class/1998/). The computation time of each algorithm is showed in Figure 4, each point represents average value of 1000 times semantic similarity computation.

Figure 4 shows that, influenced by many factors, the computation time does not remain constant, but fluctuates within a relative small range. No matter how it changes, the time consumed by our optimization algorithm is always less than that of original algorithm, and approximately 50%.

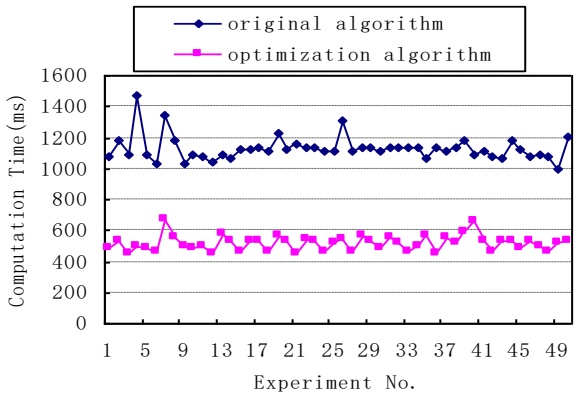

Figure 4. Similarity Computation time of Two Algorithm

Secondly, we selected 77 ontologies to evaluate the influence of ontology scale on optimization performance. These ontologies come from Protégé Ontology Library of Stanford University (http://protegewiki.stanford.edu/index.php/Protege_Ontology_Librar-y). Concepts scale range from 4633 to 8. The result is shown in Figure 5. Optimization algorithm can reduce computation complexity considerably, and computation speed was approximately improved by one time. This is consistent with the complexity analysis in section 4. Ontologies of No. 4, 11, 15~26 have fewer concepts, the optimization performance is improved under 40%.

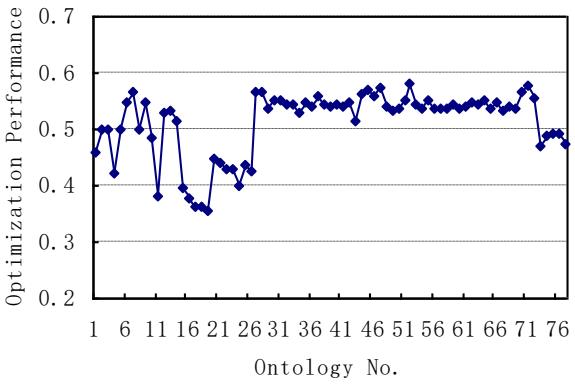

Figure 5. The Influence of Ontology Scale

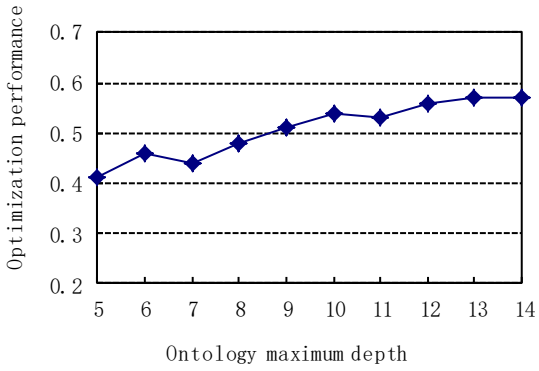

Figure 6. The Influence of Ontology Maximum Depth

Finally, we selected another 100 ontologies to evaluate the influence of ontology maximum depth on optimization performance. Ontology maximum depth range from 5 to 14. Each depth includes 10 ontologies. Figure 6 shows the results. The bigger the maximum depth of ontology, the more significantly the performance improved.

5. Conclusion

Semantic similarity computation is of great importance in many applications. There are many ontology-based semantic similarity measures were given in recent years. Semantic similarity computations are tedious and time-consuming. This paper puts forward a concept-hierarchical-relationship-based optimization algorithm to simplify semantic similarity computation. Based on the semantic similarity of one concept pair, the optimization algorithm can give semantic similarity of arbitrary concept pair in the ontology. Simulation experiments show that the computation complexity was reduced considerably, and similarity computation speed was improved by one time.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No.61402482), the Fundamental Research Funds for the Central Universities (No.2014QNB23).

References

[1] Z. Feng, F. Fei and Y. Fengwei, "Expanding approach to information retrieval using semantic similarity analysis based on WordNet and WikiPedia", International Journal of Software Engineering and Knowledge Engineering, vol. 22, no. 2, **(2012)**, pp. 305–322.
[2] A. Neda, K. Latifur and T. Bhavani, "Optimized ontology-driven query expansion using map-reduce framework to facilitate federated queries", Computer Systems Science and Engineering, vol. 27, no. 2, **(2012)**, pp. 103-115.
[3] H. Myunggwon, C. Chang and K. Pankoo, "Automatic Enrichment of Semantic Relation Network and Its Application to Word Sense Disambiguation", IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 6, **(2011)**, pp. 845-858.
[4] E. Zakaria and A. Karima, "Arabic Text Categorization: a Comparative Study of Different Representation Modes", International Arab Journal of Information Technology, vol. 9, no. 5, **(2012)**, pp. 465-470.
[5] M. Gan, X. Dou and R. Jiang, "From Ontology to Semantic Similarity: Calculation of Ontology-Based Semantic Similarity", The Scientific World Journal, vol. 2013, **(2013)**, doi:10.1155/2013/ 793091.
[6] H. Dong, F. K. Hussain and E. Chang, "A context-aware semantic similarity model for ontology environments", Concurrency computation: practice and experience, vol. 23, **(2011)**, pp. 505-524.
[7] D. Sánchez, M. Batet, D. Isern and A. Valls, "Ontology-based semantic similarity: A new feature-based approach", Expert Systems with Applications, vol. 39, no. 9, **(2012)**, pp. 7718-7728.
[8] W. Song, C. H. Li and S. C. Park, "Genetic algorithm for text clustering using ontology an evaluating the validity of various semantic similarity measures", Expert systems with applications, vol. 36, **(2009)**, pp. 9095-9104.
[9] Y. Li, Z. Bandar and D. Mclean, "An approach for measuring semantic similarity between words using multiple information sources", IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 4. **(2003)**, pp. 871-882.
[10] A. Hliaoutakis, G. Varelas, E. G. M. Petrakis and E. E. Milios, "MedSearch: A Retrieval System for Medical Information Based on Semantic Similarity", In 10th ECDL European Conference on Research and Advanced Technology for Digital Libraries, Alicante, Spain, **(2006)**, pp. 512-515.
[11] R. Rada, H. Mili, E. Bichnell and M. Blettner, "Development and application of a metric on semantic nets", IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, **(1989)**, pp. 17-30.
[12] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification", In WordNet: An Electronic Lexical Database, Fellbaum C (ed.). Wiley: New York, **(1995)**, pp. 265-283.
[13] Z. Wu and M. Palmer, "Verb semantics and lexical selection", In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, Las Cruces, New Mexico, USA, **(1994)**, pp. 133-138.
[14] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy", In Proceedings of the 14th international joint conference on artificial intelligence, Montreal, Quebec, Canada, **(1995)**, pp. 448-453.
[15] D. Lin, "An information-theoretic definition of similarity", In Fifteenth International Conference on Machine Learning, Madison, Wisconsin, USA, **(1998)**, pp. 296-304.
[16] G. Pirro, "A semantic similarity metric combining features and intrinsic information content", Data and Knowledge engineering, vol. 68, **(2009)**, pp. 1289-1308.
[17] J. Jiang and D. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy", In Proceedings of the international conference research on computational linguistics, Taiwan, **(1997)**, pp. 19-33.
[18] H. Al-Mubaid and H. A. Nguyen, "A cluster-based approach for semantic similarity in the biomedical domain", In 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, New York, USA, **(2006)**, pp. 2713-2717.

