Advances in Computer Science Research, volume 62
2nd Joint International Information Technology, Mechanical and Electronic Engineering Conference (JIMEC 2017)

# An Improved Algorithm For Conceptual Semantic Similarity In Domain Ontology

*Yaya Zhen, Xian Zhong, Lin Li, Luo Zhong*

*School of computer science and technology, Wuhan University of Technology, Wuhan 430070,China*
*zhongx@whut.edu.cn xiaohan_zhen@163.com*

**Keywords:** domain ontology; semantic similarity; node density; semantic coincidence

## Abstract

With the extensive application of ontology in many fields, the semantic similarity computation based on domain ontology has been a hot research topic. At present, taking advantage of the upper and lower layer structure of ontology to calculate the semantic similarity is the most common approach. But for these approaches, the analysis of selected factors is not comprehensive, and the result of single factor is not close to the overall semantic similarity value. In this paper, that the computing result of node density may be not in the range of meaningful value is improved, and the weight of each ancestor node is added on the basis of semantic coincidence. Considering the semantic distance, node depth, node density and semantic coincidence, an improved algorithm for conceptual semantic similarity in domain ontology is proposed. By comparing and analysing the results of the experiment, it shows that the improved algorithm has higher accuracy for the value of single factor and the overall semantic similarity.

## 1 Introduction

With the development of network technique, the network information is expanding rapidly. As the product of the new generation Internet, the emergence of the Semantic Web makes it possible to make semantic interaction between human and computer. The Semantic Web appends the semantic information that the computer can understand. It is conducive to semantic retrieval in web resources. Before the emergence of the semantic web, information retrieval is achieved only by simple matching from the perspective of grammar and words. However, it is lack of semantic description of knowledge[6]. In this case, it will not able to meet two cases, on the one hand the same concept can be expressed in many different ways, on the other hand a number of different concepts can be expressed with same expression. The conceptual semantic similarity is the key of semantic retrieval. Ontology is a conceptual model to describe the semantic information. Concept and the relationship between concepts can be described well by ontology which provides the basis for the calculation of semantic similarity.

At present, there are three basic methods to compute conceptual semantic similarity in ontology. They are based on semantic distance[4], the information content[7] and the attribute of concept[9]. Many scholars have improved the calculation of semantic similarity. Liu Ziyu[5] proposed MD4 model to calculate semantic similarity, considering the concept attribute and the semantic structure strength characteristics; Cui Qiwen[3] optimized the calculation method of concept semantic similarity in domain ontology by applying tree hierarchy of domain ontology. Batet M [1] made use of non-common ancestor node to calculate the concept semantic similarity.

The main research in this paper: 1) The situation that the calculation result of the node density is not in the range of meaningful value is improved; 2) The existed calculation methods for the influence of semantic coincidence only consider the number of common ancestor nodes of two nodes, which is some sketchy, we append the weight of each ancestor node on the basis of semantic coincidence, and improve the calculation of the semantic coincidence; 3) An improved method is proposed to calculate the semantic similarity based on the relationship between the upper and lower levels of ontology by making use of four factors, and they are semantic distance, node depth, node density and semantic coincidence respectively.

## 2 General Algorithm For Conceptual Semantic Similarity

In this paper, semantic distance, node depth, node destiny and semantic coincidence are selected to calculate semantic similarity. So we will analyse the factors of general algorithm for semantic similarity from these four aspects.

### 2.1 Semantic Distance

The kernel of this algorithm is to regard all the directed edges as 1 in domain ontology. Then, the total number of directed edges between two nodes is regarded as the semantic distance between two nodes. The computation of the semantic distance factor must satisfy that the larger semantic distance of two nodes is, the smaller semantic similarity is. Finally, we can get the Equation of the concept similarity calculation based on the semantic distance[12].

$$\mathrm{distance\_}sim(A,B) = \frac{2\times(H-1)-L}{2\times(H-1)} \quad (1)$$

In the Equation (1), H is the maximum depth of the tree structure of the domain ontology. L is the semantic distance between A and B.

### 2.2 Node Depth

In the tree structure of domain ontology, the concept whose depth is larger is the refinement of the concept whose depth is smaller, and the meaning of its expression is more specific. On the contrary, the concept whose depth is smaller is the abstract of the concept whose depth is lager, and the meaning of its expression is more abstract. Obviously, the semantic similarity between concepts whose expression is more specific is larger than concepts whose expression is more abstract. In the circumstances that the semantic distance is the same between two concept nodes, the larger depth of two nodes is in the tree structure of domain ontology, the lager semantic similarity is. Finally, we can get the Equation of the concept similarity calculation based on the node depth[1,8].

$$depth\_sim(A,B) = \frac{2 * N_{LCS}}{N_A + N_B + 2 * N_{LCS}} \quad (2)$$

In the Equation (2), $N_{LCS}$ represents the number of directed edge from the node LCS which is the least common subsumed ancestor of concept A and concept B to root. $N_A$ represents the number of directed edge between the node LCS and concept A, and $N_B$ represents the number of directed edge between the node LCS and concept B.

### 2.3 Node Destiny

In the tree structure of domain ontology, nodes are dense in some area. Then, for these nodes, the larger the degree of refinement to the parent node is, the larger semantic similarity is. Finally, we can get the Equation of the concept similarity calculation based on the node density[3]:

$$\mathrm{density}\_sim(A,B) = \frac{wid(A) + wid(B)}{2 * \max(wid(Tree))} \quad (3)$$

With Equation (3), the result of two same nodes is not determined to be 1. So there is an improved method as following [10]:

$$\mathrm{density}\_sim(A,B) = \frac{2 * wid(LCS)}{wid(A) + wid(B)} \quad (4)$$

In the Equation (3) and (4), LCS represents the least common subsumed ancestor of concept A and concept B. Wid(A) represents the number of sibling nodes (including the concept A itself), max(wid(Tree)) is maximum density in the ontology tree.

### 2.4 Semantic Coincidence

The common ancestor nodes between two concepts indicate that the two concepts have some similarities, and are the expression of commonalities between two concepts. Therefore, the number of common ancestor nodes can be used to express the similarity between two concepts in a sense. The more number of common ancestor nodes is, the greater similarity is. Finally, we can get the Equation of the concept similarity calculation based on the semantic coincidence[2]:

$$\mathrm{coincidence}\_sim(A,B) = \frac{|T_A \cap T_B|}{|T_A \cup T_B|} \quad (5)$$

In the Equation (5), $T_A$ and $T_B$ represent the set of ancestor for concept A and concept B (including itself) respectively.

## 3 Improved Algorithm For Conceptual Semantic Similarity

### 3.1 Improvement of Key Factors

#### 3.1.1 Improvement of Node Destiny

The above two Equations for calculating the node density have some disadvantages. Equation(3) only focus on the number of sibling nodes, but the influence of node density is local, but not a separate concept. And when calculating the similarity of two same concept nodes, the result is not determined to be 1. That the result is greater than 1 may happen when calculating the similarity of two concepts with Equation (4). Obviously, it is incorrect. In conclusion, for node density, the least common subsumed ancestor of two concepts needs to be considered. Finally, we can get the Equation of the concept similarity calculation based on the node density:

$$\mathrm{density}\_sim(A,B) = 1 - \frac{|2 * wid(LCS) - wid(A) - wid(B)|}{\max(wid(Tree))} \quad (6)$$

In the Equation (6), LCS represents the least common subsumed ancestor of concept A and concept B. Wid(A) represents the number of sibling nodes (including the concept A itself), max(wid(Tree)) is maximum density in the ontology tree.

#### 3.1.2 Improvement of Semantic Coincidence

In the Equation (5), the calculation of semantic coincidence only considers the number of common ancestor nodes of two concepts, which is some sketchy. In fact, the contribution of each common ancestor node to semantic coincidence of two concepts is different. The closer common ancestor node to the two concepts is, the greater contribution to the semantic coincidence of the two concepts is. Therefore, it is concluded that the semantic coincidence of the two concepts is associated with two factors. One is the number of common ancestor nodes, the other is the contribution of each common ancestor node to semantic coincidence of the two concepts, namely, the sum of the distance between the common ancestor nodes and the two concepts.

The distance between two concepts can be expressed by the depth difference of two concept nodes if a concept is the ancestor of another concept. We use depth difference to express the distance from two concepts to their ancestor node. so, if the two concept nodes are A and B, C is the common ancestor node of two concepts, the distance from two concept A, B to the common ancestor node C can be expressed as $Dep(A) + Dep(B) - 2 * Dep(C)$. As described, we can know that if the greater distance is, the smaller the contribution of semantic coincidence concept C to A and B is. Therefore, the contribution of semantic coincidence concept C to A and B is expressed as $1 - \frac{Dep(A) + Dep(B) - 2 * Dep(C)}{Dep(A) + Dep(B)}$, namely, $\frac{2 * Dep(C)}{Dep(A) + Dep(B)}$. Then, it will be done to calculate the contribution of semantic coincidence for all common ancestor nodes in the same way. Finally, the sum of the contribution values of all common nodes is obtained.

In addition to common ancestor nodes, non-common ancestor nodes also have an impact on the similarity of the two concepts. The more the number of non-common ancestor nodes, the smaller similarity between two concepts. The union of ancestor nodes of the two concepts can be divided into two parts, one is the common ancestor nodes, another is the non-common ancestor nodes. Nevertheless, the non-common ancestor nodes also can be divided into two parts, one is the concept node that is the ancestor node of concept A but not concept B, another is the concept node that is the ancestor node of concept B but not concept A. Making use of two kinds of non-common ancestor nodes to calculate the contribution of semantic coincidence for concept A and concept B respectively also satisfies the condition that the more distance is, the smaller the contribution value is. As mentioned above, it can be concluded that the contribution of node which is the ancestor node of concept A for concept A can be expressed as

$$1 - \frac{Dep(A) - Dep(C_A)}{Dep(A)}, \text{ namely, } \frac{Dep(C_A)}{Dep(A)} .$$

Therefore, the contribution of node which is the ancestor node of concept B for concept B is described with $\frac{Dep(C_B)}{Dep(B)}$.

Finally, according to the above analysis, we can get the Equation of the concept similarity calculation based on the semantic coincidence degree:

$$\text{coincidence\_}sim(A,B) = \frac{\sum_{C \in T_A \cap T_B} F_{A,B}(C)}{\sum_{C \in T_A \cap T_B} F_{A,B}(C) + \sum_{C_A \in T_A - T_B} F_A(C_A) + \sum_{C_B \in T_B - T_A} F_B(C_B)} \quad (7)$$

$$F_{A,B}(C) = \frac{2 * Dep(C)}{Dep(A) + Dep(B)} \quad (8)$$

$$F_A(C_A) = \frac{Dep(C_A)}{Dep(A)} \quad (9)$$

$$F_B(C_B) = \frac{Dep(C_B)}{Dep(B)} \quad (10)$$

In the above Equation (7), (8), (9) and (10), $T_A$ and $T_B$ are the set of ancestor nodes of concept A and concept B respectively (including the concept itself). $F_{A,B}(C)$ expresses the contribution of semantic coincidence of common ancestor nodes for concept A and concept B. $T_A$-$T_B$ represents the set of nodes that $T_A$ contains but $T_B$ not. $F_A(C_A)$ is the contribution of node which is in the set $T_A$-$T_B$ for concept A.

### 3.2 Improved Algorithm For Conceptual Semantic Similarity

According to the above analysis, we can get the Equation of the conceptual semantic similarity:

$$\begin{aligned} &\text{semantic\_}sim(A,B) \\ &= \alpha \bullet \text{distance\_}sim(A,B) + \beta \bullet depth\_sim(A,B) \\ &\quad + \gamma \bullet \text{density\_}sim(A,B) + \varphi \bullet \text{coincidence\_}sim(A,B) \end{aligned} \quad (11)$$

In the Equation (11), $\alpha, \beta, \gamma, \varphi$ are the weight of impact for semantic distance, node depth, node density and semantic coincidence respectively, and the range of value is [0, 1]. $\alpha + \beta + \gamma + \varphi = 1$ must be satisfied by the parameters. By adjusting the four parameters, it can be more reasonable in different domain to calculate semantic similarity. These four parameters are empirical value, and the selection of specific value is seen by reference to the experimental part in this paper.

## 4 Experimental Results Analysis

### 4.1 Experimental Environment

The experiment of the algorithm for conceptual semantic similarity is achieved with java. Software environment: operating system Windows7, development platform Eclipse, ontology modelling tool Protege4.3, ontology library interface ProtegeAPI. The experimental data: It uses the crop ontology example presented in the references [11] to compare with the results of references [11] for convenience. The crop ontology is used to test the effect of semantic similarity in many papers. The crop ontology example is shown in Figure 1. In this experiment, according to the empirical value to set adjustable parameters, $\alpha, \beta, \gamma, \varphi$ are 0.1, 0.04, 0.06, 0.8 respectively.

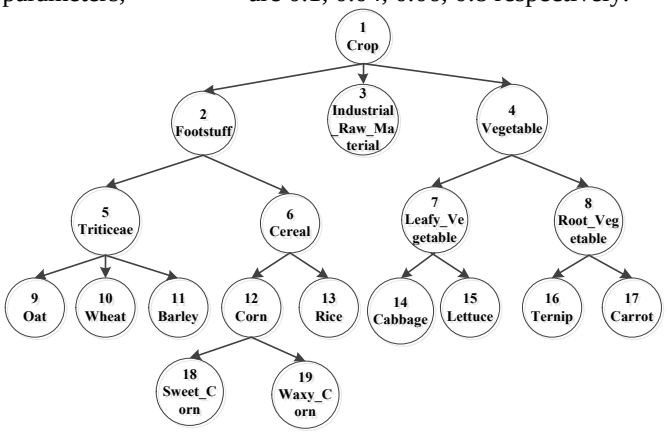


Figure 1: Crop ontology example

### 4.2 Experimental Results

In this paper, the algorithm combines four factors that are semantic distance, node depth, node density and semantic coincidence, and improves the calculation of node density and semantic coincidence these two factors. The results of the experiment and the results of the two classical algorithms are compared with the expert's experience value which is a weighted mean derived from a number of value given by a number of experts in related areas.

The verify mode of this experiment as following: The closer the value of key factor is to expert's experience value, the more effective algorithm is; the closer the value of overall semantic similarity is to expert's experience value, the more effective algorithm is.

#### 4.2.1 Comparative Analysis of Experimental Results for Improved Key Factors

**(1) comparative analysis of node density**

That the results of the node density calculation and the results of Equation (3) and (4) are compared with the expert's

experience value is shown in Figure 2. The Equation (3) and (4) are used to calculate node density in many kinds of semantic similarity algorithm. The abscissa in the graph indicates the semantic similarity between the corresponding serial numbers. For example, 10/13 indicates semantic similarity between Wheat and Rice.

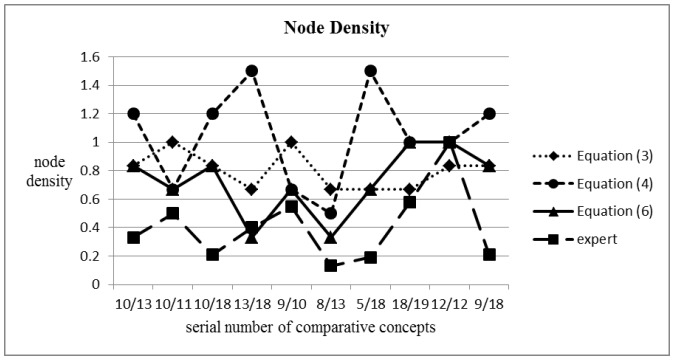


Figure 2: Experimental results of node density

From the Figure 2, we can see that compared with the Equation (3), although the results are close to each other, the Equation (3) cannot ensure the result is always 1 when the node density is calculated for the same two nodes. In this paper, the algorithm is able to ensure that the result of every factor is always 1 for the same nodes. Compared with the Equation (4), the results are not only closer to expert's experience value, but also solve the problem in Equation (4) that the result may be greater than 1, namely, ensure the value in the range of meaningful value[0,1].

**(2) comparative analysis of semantic coincidence**

That the results of the semantic coincidence calculation and the results of Equation (5) are compared with the expert's experience value is shown in Figure 3. The Equation (5) is used to calculate semantic coincidence in many kinds of semantic similarity algorithm.

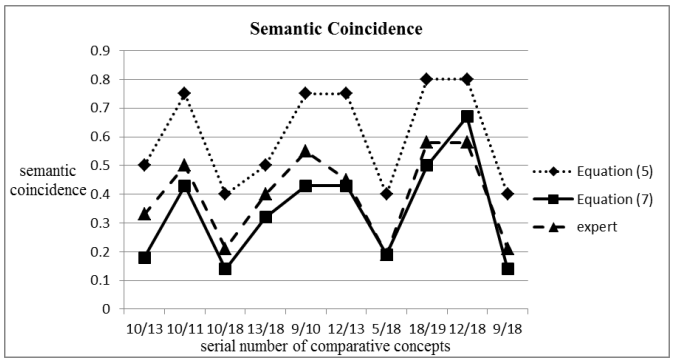


Figure 3: Experimental results of semantic coincidence

From the Figure 3, we can see that compared with the Equation (5), the results are closer to expert's experience value. So it shows that the improved algorithm of semantic coincidence calculation is better.

### 4.2.2 Comparative Analysis of Experimental Results for improved Semantic Similarity Algorithm

That the results of the semantic similarity and the results of the two classical algorithms are compared with the expert's experience value is shown in Table 1 and Figure 4. The first algorithm comes from the references [13], and the second algorithm comes from the references [11].

| Serial number | Comparative concepts | algorithm 1 | algorithm 2 | Equation （11） | expert |
|---|---|---|---|---|---|
| 10/13 | Wheat Rice | 0.26 | 0.52 | 0.26 | 0.33 |
| 10/11 | Wheat Barley | 0.33 | 0.53 | 0.49 | 0.5 |
| 10/18 | Wheat Sweet_corn | 0.19 | 0.46 | 0.21 | 0.21 |
| 13/18 | Rice Sweet_corn | 0.27 | 0.54 | 0.32 | 0.4 |
| 9/10 | Oat Wheat | 0.4 | 0.57 | 0.49 | 0.55 |
| 8/13 | Root_Vegetable Rice | 0.08 | 0.41 | 0.07 | 0.13 |
| 5/18 | Triticeae Sweet_corn | 0.16 | 0.44 | 0.25 | 0.19 |
| 18/19 | Sweet_corn Waxy_corn | 0.37 | 0.58 | 0.57 | 0.58 |
| 12/13 | Corn Rice | 0.33 | 0.54 | 0.49 | 0.45 |
| 9/18 | Oat Sweet_corn | 0.19 | 0.46 | 0.21 | 0.21 |

Table 1: Experimental results of Semantic similarity

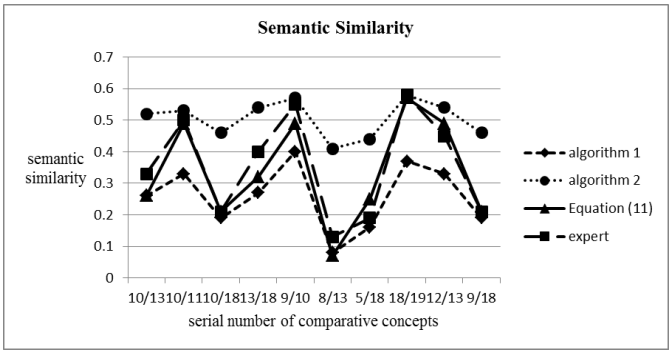


Figure 4: Experimental results of Semantic similarity

From the Table 1 and Figure 4, we can see that compared with the algorithm 1 and algorithm 2, the results are closer to expert's experience value. So it shows that the improved algorithm of semantic similarity is better.

According to the above analysis, we can conclude that the improved algorithm for conceptual semantic similarity satisfies the following characteristics:

(1) The algorithm not only guarantees the overall semantic similarity value is in the range of meaningful value[0, 1], but also ensure every factor value is in [0, 1];

(2) For the same nodes, the algorithm not only guarantees the overall semantic similarity value is always 1, but also ensure every factor value is always 1;

(3) Compared with the original classical algorithms at present, the values have higher accuracy, and the results are closer to the expert's experience value.

#### 4.2.3 Accuracy Measurement

In order to verify the accuracy of the algorithm in this paper, we use the vector Euclidean distance to measure the accuracy of the results. Here, the results of calculated by each algorithm are expressed as a vector. For example, the expert's experience value vector can be expressed as $V_{exp}$={0.33, 0.5, 0.21, 0.4, 0.55, 0.13, 0.19, 0.58, 0.45, 0.21}. In this paper, we need calculate Euclidean distance between expert's experience value and node density, semantic coincidence, semantic similarity to verify the accuracy of the algorithm. The smaller Euclidean distance is, the better algorithm is. The results of Euclidean distance between expert's experience value and the following elements are shown as Table 2.

| factors | Comparative elements | Euclidean distance |
|---|---|---|
| Node density | Equation (3) | 1.451 |
| | Equation (4) | 2.449 |
| | Equation (6) | 1.232 |
| semantic coincidence | Equation (5) | 0.667 |
| | Equation (7) | 0.270 |
| semantic similarity | algorithm 1 | 0.342 |
| | algorithm 2 | 0.579 |
| | Equation (11) | 0.162 |

Table 2: The results of Euclidean distance

From the Table 2, we can see that :
(1) Compared with Equation (3) and Equation (4), the Euclidean distance between expert's experience value and Equation (6) is the smallest. So, the node density in this paper has higher accuracy than Equation (3) and Equation (4).
(2) Compared with Equation (5), the Euclidean distance between expert's experience value and Equation (7) is smaller. So, the semantic coincidence in this paper has higher accuracy than Equation (5).
(3) Compared with algorithm 1and algorithm 2, the Euclidean distance between expert's experience value and Equation (11) is the smallest. So, the semantic similarity in this paper has higher accuracy than algorithm 1 and algorithm 2.

## 5 Conclusion and future work

In this paper, based on the upper and lower layer structure of the ontology tree, it improves the calculation of node density and semantic coincidence, and gets semantic similarity value which is a weighted mean derived from semantic distance, node depth, node density and semantic coincidence. Compared with the original classical algorithm, the accuracy of the semantic similarity in this paper has been improved obviously. In this paper, the results are improved according to the hierarchical structure of ontology. The influence factors of the non-hierarchical structure can be further considered. Finally, the target that the calculation method is simple and the influence factors considered of ontology semantic are comprehensive will be achieved.

## Acknowledgements

This work was supported by Project Supported by Project No.2015BAA072 , No.2015CFB525 and No.61003130.

## References

[1] Batet M, Sánchez D, Valls A. An ontology-based measure to compute semantic similarity in biomedicine.[J]. Journal of Biomedical Informatics, 2011, 44(1):118-125.
[2] CAO Rui, WU Lin-da. An Improved Semantic Similarity Algorithm Bade on Domain ontology [J]. Microelectronics & Computer, 2014(8): 109-114.
[3] CUI Qi-wen, XIE Fu. Improved Computational Method for Conceptual Semantic Similarity in Domain Ontology[J]. Computer applications and software, 2012, 29(2): 173-174.
[4] Fellbaum C, Miller G. An Electronic Lexical Database[J]. Cognition Brain & Behavior, 1998:265-283.
[5] LIU Zi-yu. Research on Construction and Semantic Retrieval of Multiple Majors Domain Ontology[D]. Beijing Jiaotong University, 2011, 2009.
[6] Liu Z, Zhang Y. Research and Design of E-commerce Semantic Search[C]. IEEE 3rd International Conference on Information Management, Innovation Management and Industrial Engineering. 2010:332-334.
[7] Resnik P. Semantic Similarity in a Taxonomy: An Information-based Measure and its Application to Problems of Ambiguity in Natural Language[J]. Journal of Artificial Intelligence Research, 2011, 11(1):95-130.
[8] Sánchez D, Solé-Ribalta A, Batet M, et al. Enabling Semantic Similarity Estimation Across Multiple Ontologies: An evaluation in the biomedical domain[J]. Journal of Biomedical Informatics, 2012, 45(1):141-155.
[9] Tversky A. Features of Similarity[J]. Readings in Cognitive Science, 1988, 84(4):290-302.
[10] WANF Zhen, LU Neng-zhi. Improvement of Semantic Similarity Algorithm Based on Tree Structure[J]. Modern computer, 2015(6): 27-30.
[11] ZHANG Lan-fang. Natural Language Semantic Similarity Algorithm Based on Ontology [J]. Journal of Guilin University of Technology, 2012, 32(2): 253-258.
[12] ZHAO Peng-wei, YUAN Ying. Research on Semantic Similarity Computing Methods Based on Domain-Ontology[J]. Sci-Tech Information Development & Economy, 2010,20(8):74-77.
[13] ZHAO Yong-jin, ZHENG Hong-yuan, DING Qiu-lin. Study on Semantic Similarity Algorithm Based on Ontology[J]. Journal of Computer Applications, 2009,29(11):3074-3076.