Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2013, 5(9):324-332

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

A similarity computing method based on a domain ontology for forest pests and diseases

Dongmei Li[1,2], Na Li[1], Jiajia Hou[1], Qin Mo[1] and Junxiang Wang[1]

[1]*School of Information Science and Technology, Beijing Forestry University, Beijing, China*
[2]*School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China*

ABSTRACT

The domain knowledge of forest pests and diseases refers to many different disciplines with large numbers of knowledge .It is a significant and challenging project to share and reuse this knowledge and to retrieve what you need intelligently and efficiently. In this paper, ontology as the modeling approach is introduced into the field of forest protection .Through studying the characteristics of the knowledge of forest pests and diseases, this paper constructs a domain ontology to obtain concept semantic tree as the foundation of computing semantic similarity. The novel approach computes semantic similarity according to hierarchical structures and inheritance relationships in the concept semantic tree. The automatic question answering system from this model is able to accept natural language description, and return relevant answers automatically after word segmentation, semantic similarity computing and statements similarity computing. Experimental results show high accuracy and integrality achieved by using this retrieval system.

Key words: Similarity Computing, Concept Semantic Tree, Domain Ontology, Forest Pests and Diseases.

INTRODUCTION

Effective management of forestry knowledge is the basis of achieving forestry informationization [1]. With the purpose of sharing and reusing domain knowledge of forest pests and diseases, forestry management experts have been exploring an effective, intelligent method to reveal the complex relationship about the knowledge in the field of forest pests and diseases, and search an efficient way to deal with it. In [2] and [3], the authors not only analyze the diagnosis and treatment processing of forest diseases and insect pests, the composition of forest diseases and pests knowledge as well as the forms of expression and representation of the knowledge, but also design the knowledge base of forest pests and diseases and the corresponding reasoning mechanism. As a result, the diagnosis and treatment of forest pests and diseases system is implemented. However, this traditional expert system gets answers to users' questions by reasoning knowledge stored in the knowledge base. The knowledge base and reasoning mechanism are merely designed for a particular system and difficult to be shared and reused with other expert systems in the related field. As a conceptual modeling tool describing information system in semantic and knowledge level, through sharing concepts, ontology can provide a common understanding about knowledge in different areas, and at the same time mutual understanding of the semantics between man and machine, machine and machine can be achieved. With the introduction of ontology to the field of forestry knowledge management, the problems of the traditional knowledge representation methods are effectively solved.

In [4], Fan etc. propose an ontology-based method for forest channel knowledge management. This method helps make strategic decisions in forest channel knowledge management, but it does not involve the field of knowledge management of forest diseases and insect pests. In [5], Wang etc. present the formal definition of forest disease and pest diagnosis ontology, which can reveal the clear hierarchy structure of knowledge. However, this paper focuses on ontology reasoning and conflict detection based on description logic and doesn't accept natural language description of the problem. Based on [4] and [5], this paper introduces ontology into the forest protection field. A domain ontology model constructed is used to improve the concept similarity algorithm. Finally his paper implements a retrieval system based on the concept semantic tree. The system can accept two different input methods, including word query and statement query. If the user input a query request into the system in the form of a word or statement in natural language, the system can return relevant answers automatically from ontology database and web after word segmentation, semantic similarity computing and statements similarity computing. Quantitative calculation is used to avoid the problems occurring in the process of dealing with traditional natural language. At the same, recall and precision are largely improved.

EXPERIMENTAL SECTION

The analysis and design of the domain ontology

The field of forest pests and diseases is mainly a collection of concepts, and it involves a large breadth of knowledge in the view of sub-concepts of this field. Thus proceeding from three fields of China forest, trees, pests, with semantic matching ability of the ontology, our system solves the issue of the similarity or relatedness of concepts and determines the three top-level concepts, which are Chinese forest partition ranging from district one to the district two, major trees family and major forest pests. In addition, major tree family is distinguished by the number of genera and the main forest pests are departed according to the disease-prone areas. In this basis, our system improves the current forest pests and diseases system, and defines the concept property as well as creates instances to build the product ontology.

Representation of the domain ontology

In this paper, ontology standard language OWL is selected as the domain ontology description language, along with Protégé software version 4.1 as editing tool. Combined with previous work, this paper inputs the concept hierarchical graph of forest pests and diseases into Protégé. With the help of Graphviz's plugin, the hierarchy relationships between forest pest domain ontology concepts are formally showed.

There are three fundamental objects of OWL ontology which are Class, Individual, and Property. Being the most basic concept in ontology field, OWL class is a collection of individuals and corresponds to the root in the field hierarchical tree.

```
<owl:Classrdf:ID="Northeast China">
<rdfs:subClassOf>
<owl:Classrdf:ID=" Sanjiang Plain "/>
</rdfs:subClassOf>
</owl:Class>
```

From the OWL code above, the hierarchical structure and inheritance relationship of classes are obvious which illustrates the northeast China is a regional part of Sanjiang plain. Reflected in the OWL language, their relationship is *Sub Class Of.* Furthermore, there are three other kinds of relationships between classes, *Disjoint Classes, Sub Class Of, Equivallent Classes are,* which are three axioms, and respectively represent subclasses, different classes and similar classes. In terms of that a class is used to describe the common attributes of some individuals, an individual should be stated as a class member when being introduced.

OWL uses class to describe all the individuals' common properties of the class, and the individual is the one that can actually be used or that are important. It simply needs to declare it as a member of a class when introducing an individual.

According to above, semantic relations between these concepts play an important role in improving the accuracy and integrity rate of search results. But how can the system present the complicated relationships between class and class,

class and example, example and example? This paper uses the OWL attribute to solve this problem and fully defines these relations. In fact, every attribute is a binary relation.

<owl:Classrdf:ID=" Tung oil tree inchworm ">
<rdfs:subClassOf>
<owl:Restriction>
<owl:someValuesFrom>
<owl:Classrdf:about="# Taxodiaceae "/>
</owl:someValuesFrom>
<owl:onProperty>
<owl:ObjectPropertyrdf:about="# Hazard trees have "/>
</owl:onProperty>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

From this example it can be found that the way of defining the attribute "harm trees have" in the domain ontology is *someValuesFrom*. In other words, Tung oil tree inchworm hazards some Taxodiaceae species but not all Taxodiaceae species. In addition, the ontology in this paper also sets a mutually inverse relationship. For instance, if attribute P1 is marked as inverse of attribute P2, then for all the x and y, they satisfy P1 (x, y) only when P2 (y, x). As shown in Figure 1, it defines six properties. According to the description of the object and the role of the object, any two properties have a mutually inverse relationship. For example, northeast China forest area has pinaceae, therefore northeast China is listed in the pinaceae trees distribution graph.

The construction of domain ontology forest diseases and insect pests is all conducted in Protégé as demonstrated in Figure 1. Moreover, it does not show obvious errors when being taken the ontology effectiveness, consistency and conflict tests by Jena, which indicates that this ontology construction is effective and constructive.

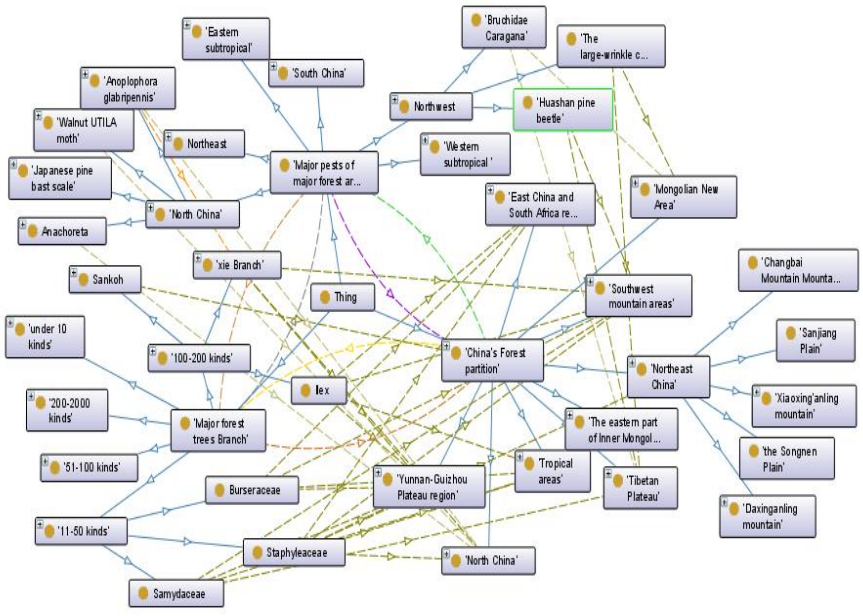

Figure 1. *Forest Diseases and Pests Domain Ontology*

Word segmentation algorithm

Owing to the fact that the proper nouns of forest pests and diseases are long and building questions sentence library is very difficult, this paper improves the existing word segmentation algorithms and adopt the forward largest word segmentation algorithm based on the dictionary [6]. This paper regards concepts involved in the ontology as a word segmentation dictionary, and carries the queries which users put forward in natural language on Chinese word

segmentation, part-of-speech tagging, segmentation annotation, along with the additional function of processing simple statements through word segmentation algorithm(Algorithm descriptions are shown in algorithm 1,2).

Algorithm 1：// Algorithm for finding the biggest word segmentation
Input：Word segmentation dictionary files
Output：The maximum word segmentation length
Load(Word segmentation Dictionary) // Load word segmentation dictionary files
// Search word segmentation dictionary files
N = 0; Biglength = Word segmentation Dictionary[N];
While not eof(Word segmentation Dictionary) do
// Deal with the N-th concept in the word segmentation dictionary
For each of Word segmentation Dictionary[N] do
If Word segmentation Dictionary[N].length > Biglength
Biglength = Biglength
Output(Biglength)

Algorithm 2：// Algorithm for handling user's request statements for word segmentation
Input：user's request statements(S1)
Output：word segmentation statement vector(V)
Init(S1), Load(S1)// Initialize and load the user's request statements
Load(Word segmentation Dictionary)// Load word segmentation dictionary files
Load(Biglength)//Load the maximum word segmentation length
Startpointer—>S1 Starting position; Movepointer->S1 Starting position
While not eof(S1) do
While not (Movepointer-Startpointer).length > Biglength
//Deal with the contents between the two pointers
S=Contents between pointers Startpointer and Movepointer
If(S go Word segmentation dictionary matching the same contents succeeds)
S InsertTo(V); Startpointer = Movepointer+1; Movepointer = Movepointer+1;
Else
Movepointer++;
Output(V)

For example, if users input a statement: "What places are pines living in ", after processing this sentence it can get a keyword sequence with part-of-speech tagging: " what/ i places / n are/ u pines / n living in/ v ". Due to the fact that any sentences consist of the key components (subject, predicate, objective, etc.) and modified components (attribute, adverbial, complement, etc.), the key components play a major role in the sentence. Therefore, we only need to consider the key components of the sentence to form the keyword sequence "What places are pines living in ". Meanwhile, dictionary in this paper is along with annotation. Annotating the keywords is required after getting the keyword sequence. "Pine" in the dictionary is an annotation of "pine family", and "living" is an annotation of "trees distribute in ". Words behind in the sentence are all concepts or attributes in the ontology. It will get respective annotations in the segmentation module, and transform them into concept or attribute words to facilitate the realization of the late similarity algorithm.

Conceptual similarity algorithm

Ontology uses the structure of hierarchical tree to describe the logical relationship which provides basis for retrieval algorithm [7-11]. This paper improves the concept similarity algorithm proposed by Wang Jin. It considers concepts semantic relations, the hierarchical structure and inheritance relationship and other factors, processes different types of relationships between concept tree grandparent and grandchild nodes and sibling nodes combined with the similarity and correlation between different concepts, and conducts a description and quantization to the concept similarity to improve retrieval accuracy [12]. This paper makes the following definitions to calculate the concept similarity.

Definition 1: if concept A is concept B’s ancestor in the hierarchy tree of ontology concept, then name A and B as

Same Branch Concepts, denoted as *S (A, B)*. Concept A is called the closest root concept of A and B, which is denoted as *R (A, B)*. The distance between A and B, *d (A, B)*, is equal to subtraction of *dep(B)* and *dep(A)*(*d (A, B) = dep(B) - dep(A)*), where *dep (C)* represents the depth of concept C in the hierarchical structure.

Definition 2: if concept A is not concept B's ancestor and concept B is not concept A's ancestor in the hierarchy tree of ontology concept, then name A and B as *Different Branch concepts*, denoted as *D (A, B)*. If concept R is the common ancestor of both A and B, and is the farthest node to root node among all nodes meeting with the condition above, then R is called as the latest root concept of A and B, which is denoted as *R (A, B)*. The distance between A and B, *d (A, B)*, is equal to the addition of *d (A, R)* and *d (B, R)* (*d (A, B) = dep(B) + dep(A)*).

Therefore, there are only three types of relation between any two concepts' in ontology concept tree that are *Same Branch concepts*, *Different Branch concepts* and *the Same concepts*. Moreover, the distance between two concepts *d (A, B)* is regarded as the length of the shortest path connecting two concepts. The greater the distance of two concepts is, the lower similarity they have. Conversely, the smaller the distance of the two concepts is, the higher similarity they have. In particular, when the semantic distance of the two concepts is zero, the similarity is one, and when the semantic distance of the two concepts is infinity, the similarity is zero.

Definition 3: in concept tree, the similarity calculation results are influenced by the number of A and B's son concepts and the amount of their semantic concepts. When A and B are the *Same Branch concepts*, named as *S (A, B)*, A is the latest concept to root of A and B, which is denoted as *R (A, B)*. Hence, R's son concept consists of B's son concepts and the semantic correlation concepts of A with B. The bigger the proportion of the latter is, the smaller the correlation between A and B is. When A and B are *Different Branch concepts*, named as *D (A, B)*, *R (A, B)* is the latest concept to root A and B. R's son concepts consist of A's son concepts, B's son concepts and semantic correlation concepts. Similarly of A and B. The bigger the proportion of the third is, the smaller the correlation of A and B is. *Son (C)* represents the number of C's son concepts.

The concept similarity is defined as in (1) according to the three definitions:

$$\mathrm{sim}(A,B)=\begin{cases} \text{when } d(A,B)\neq 0, S(A,B), \\ \left(1-\frac{\alpha}{dep(R(A,B)+1)}\right)\times\frac{\beta}{d(A,B)}\times\frac{son(B)}{son(A)} \\ \text{when } d(A,B)\neq 0, D(A,B), \\ \left(1-\frac{\alpha}{dep(R(A,B)+1)}\right)\times\frac{\beta}{d(A,B)}\times\frac{son(A)+son(B)}{son(R)} \\ \text{when } d(A,B)=0, \\ 1 \end{cases} \tag{1}$$

However, the results calculated by this approach are the same in *Same Branch* and *Different Branch concepts*, not leading to obvious differences. At the same time, distance, depth and the son concepts cannot contact closely with each other. Hence, this paper puts forward definition four.

Definition 4: the depth of ontology's position will influence the similarity calculation but the depth is relative. It can be expressed by depth formula as in (2):

$$\alpha(A,\ B)=\begin{cases} \frac{dep(A)}{dep(A)+dep(B)}, & dep(A)\leq dep(B) \\ 1-\frac{dep(A)}{dep(A)+dep(B)}, & dep(A)>dep(B) \end{cases} \tag{2}$$

In summary, this paper defines the same semantic similarity of the concept as one and improves formula(1) combined with the concept of semantic relationships, hierarichical structure, inheritable relationships and other factors to definition concept similarity of A and B(as in (3)). β and γ are adjustment factors; *d (A, B)* is the distance of A and B; *dep (C)* is the depth of C; *son (C)* is the son concept of C; α(A, B)is the depth relationship between A and B.

$$sim(A,B)=\begin{cases}\text{when } d(A,B)\neq 0, S(A,B) \text{ and } dep(A)<dep(B),\\ \frac{\beta}{1+d(A,B)}\times\gamma\frac{dep(R)}{dep(R)+\alpha(A,B)\cdot(son(A)-son(B))+(1-\alpha(A,B))\cdot son(B)}\\ \text{when } d(A,B)\neq 0, D(A,B),\\ \frac{\beta}{1+d(A,B)}\times\gamma\frac{dep(R)}{dep(R)+\alpha(A,B)\cdot son(A)+(1-\alpha(A,B))\cdot son(B)}\\ \text{when } d(A,B)=0,\\ 1\end{cases} \quad (3)$$

Statements similarity algorithm

Sentence X represents users' questions and sentence Y represents sentences in the answer library. Sentences X consists of word X_1, X_2,⋯, X_n, sentence Y consists of word Y_1, Y_2, ⋯, Y_m. The semantic similarity between all the words in sentence X and sentence Y is computed through similarity matrix which is denoted as M_{xy} [13-14].

(1)Ontology vector of users' questions: $\mathbf{X}=(x_1,x_2,x_3,\cdots x_n)$

Ontology vector of candidate sentences: $\mathbf{Y}=(y_1,y_2,y_3,\cdots y_n)$

(2)Struct the similarity matrix M_{XY} of X and Y (as in 4)

$$M_{xy}=\begin{bmatrix}sim(x_1,y_1) & \cdots & sim(x_1,y_m)\\ \vdots & \ddots & \vdots\\ sim(x_n,y_1) & \cdots & sim(x_n,y_m)\end{bmatrix} \quad (4)$$

Where $sim(x_i,y_j)$ represents the similarity of concept X_i and Y_j. Each row in the matrix represents the concept similarity between a certain word in sentence X and all the words in sentence Y.

(3)What we should do is to seek the semantic similarity between all the words in sentence X and Y to reduce the dimension. The method is to get the maximum value of each row in the matrix, which means seeking the maximum concept similarity between a certain word in sentence X and all the words in sentence Y, to compress the matrix to one-dimensional, and then to get the average value of all these maximum values, which is the semantic similarity between X and Y (as in 5).

$$Sim1=\frac{1}{n}*\sum_{i-1}^{n}(\max(X_i,Y_i),j\in[i,n]) \quad (5)$$

RESULTS AND DISCUSSION

The purpose of applying the ontology to the field of forest pests and diseases is to improve the recall rate and precision rate of the whole system, to make the system more intelligent and to make the query results more in line with the needs of users. In view of the method above, this paper implements a retrieval system of forest pests and diseases based on a domain ontology (As shown in Figure 2).

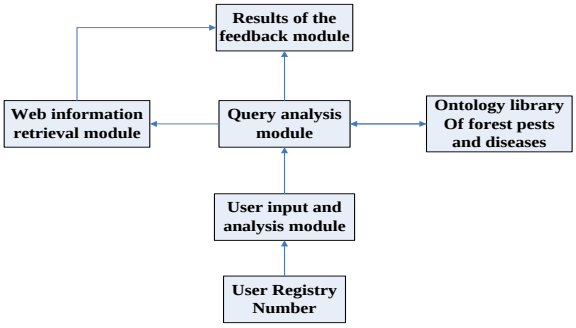

Figure 2. *Retrieval System in the Field of Forest Pests and Diseases*

The system uses the B / S structure. It submits users' queries to the server through the user login module, proceeds word segmentation processing by special user input and analysis module, and transmits the results to the query analysis module. The module calls domain ontology library on semantic expansion of the query requests, and then delivers the extended query requests to Web information retrieval module which calls Google Ajax Search API to retrieve the Google database. Besides, it optimizes and sorts the query results via the feedback module of results and delivers them to users. As a result, this paper implements the entire search process.

(1)Word Retrieval

When the system starts word retrieval, if users input ontology concepts, it extends synonymous concepts, parent concepts, sub concepts, brother concepts of the ontology. For example, if the user enters the keyword "pine family", it will show it's child-parent concepts, related concepts and concepts connected with attributes, as well as local and web search results. Local search results are the detailed concepts in the ontology library (as shown in Figure 3) and Web search results are the results regarding the search words as keywords and searching the Google database through calling Google Ajax Search API(as shown in Figure 4).

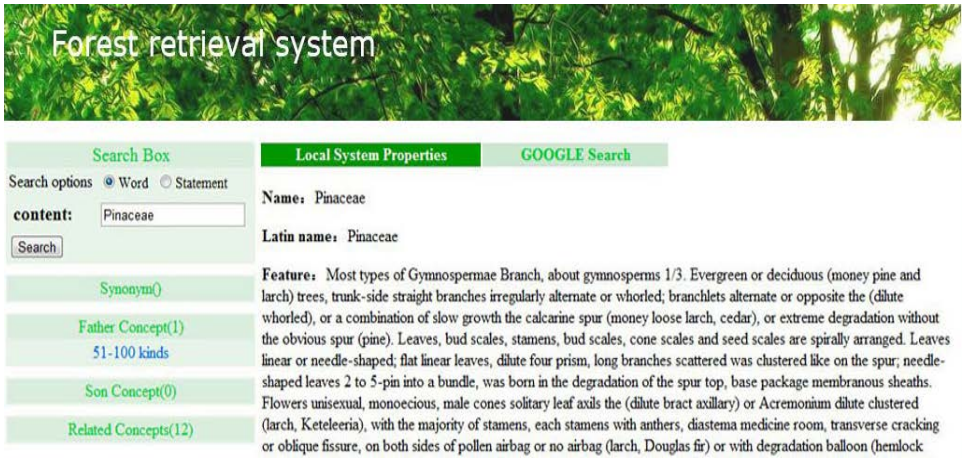

Figure 3. *Local Search Results of the Word Retrieval*

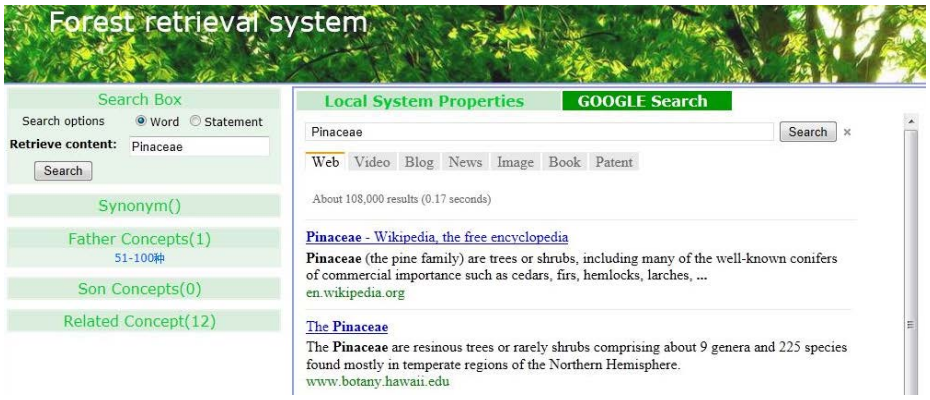

Figure 4. *Web Search Results of the Word Retrieval*

(2)Statement Retrieval

If users input in a natural language, the system proceeds word segmentation processing to the sentence, removes stop words, conducts part-of-speech tagging, extracts key parts, and annotates the words .At the same time, it gets relationships between words based on syntactic analysis, matches with the concepts and attributes in the ontology as well, and then outputs local and web search results whose sentence similarity is greater than the sentence similarity threshold users set themselves. For example, this paper sets the threshold to 0.7. If users input "what insects are pines afraid of", it will show statements whose similarity value is greater than the threshold. As shown in Figure 5 and Figure 6.

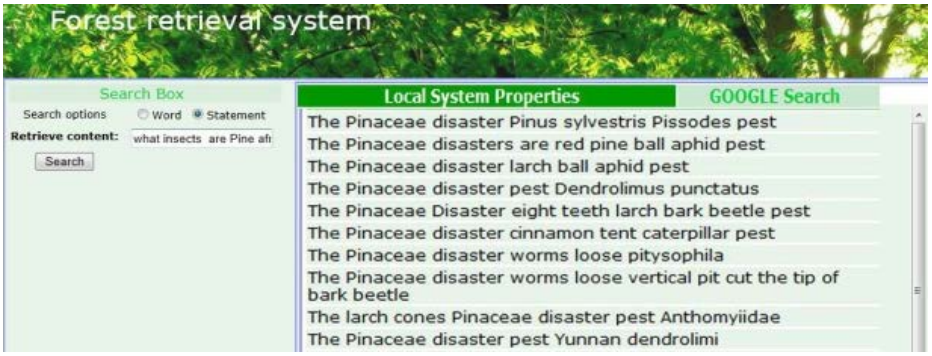

Figure 5. *Local Search Results of the Statement Retrieval*

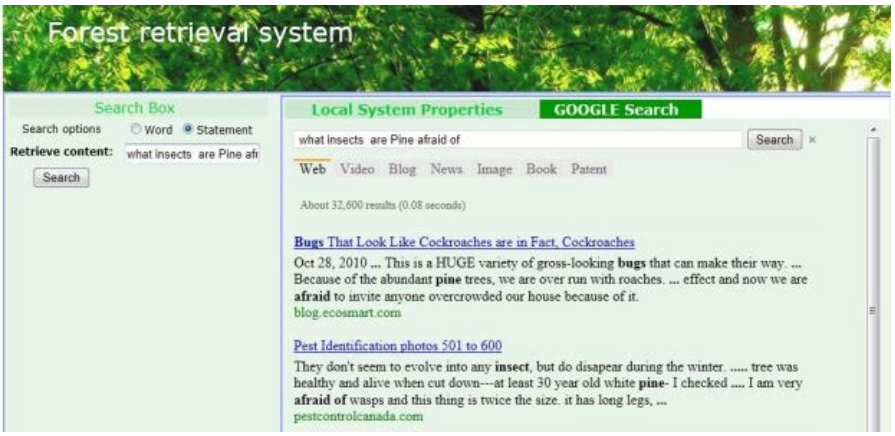

Figure 6. *Web Search Results of the Statement Retrieval*

It can be seen obviously from the operating results schematic diagram that when users input words, the system can effectively improve the recall rate of the retrieval and that when users input statements. Not only the recall rate is guaranteed, but also the system can understand users' meaning to a certain extent and have the basic intelligence and significantly enhance the recall rate. Meanwhile, when the system is running, it is very stable and fast and has the capacity of handling exceptions as well.

CONCLUSION

Aimed at semantic retrieval problems in the field of forest pests and diseases, this paper explores the basic methods of ontology used in the application of the field of knowledge representation of forest pests and diseases, and constructs a forest pests and diseases knowledge reasoning query application system based on the ontology. If the user input a retrieval request to the system in a natural language, the system obtains the semantic information associated with users' retrieval from the ontology database and web after word segmentation, semantic similarity and statements similarity computation. The system implements knowledge reasoning and inquiry, and has a certain ability to discover hidden knowledge of semantic information. Moreover, the system tests and verifies the feasibility of ontology in the application of the field of knowledge representation of forest pests and diseases which riches intelligent knowledge management research in the field of forest pests and diseases. The next step of this paper is to perfect domain ontology of forest pests and diseases, and to improve the function of the system's reasoning. We need to do more to combine the best features of ontology technology and natural language processing technology so as to get more effective search results.

ACKNOWLEDGMENT

This work was supported in part by the National College Students' Training Programs of Innovation and Entrepreneurship(No.201210022055), and the Fundamental Research Funds for the Central Universities(No.YX2011-30).

REFERENCES

[1] S.T.Kaloudisa; C.P.Yialourisb; N.A.Lorentzosb; et al. Computers and Electronics in Agriculture, **2010**, 70(2), 285-291.
[2] Baoguo Wu; Liangbao Wen. Beijing Forestry University Journal, **2006**, 28(6), 113-118.
[3] S.Kaloudisa; D.Anastopoulosa; C.P.Yialourisb; et al. Expert Systems with Applications, **2005**, 28(3), 445-452.
[4] Jing Fan; Xiuying Liu; Ying Shen; et al. In 9th International Conference on channel Fuzzy Systems and Knowledge Discovery (FSKD), **2012**, 1523-1527.
[5] NihongWang; DanLi; HuaPan; ZhiqiangLiang. In proceedings of Second International Workshop on Education Technology and Computer Science, **2010**, 416-419.
[6] Guobin He; Jinglu Zhao. Computer Engineering and Applications, **2010**, 46(3), 125-127.
[7] James Z.Wang; Zhidian Du; RapeepornPayattakool. Original Paper, **2007**, 23(1), 1274-1281.
[8] Reshadat; Vahideh; Feizi-Derakhshi; Mohammad-Reza. Research Journal of Applied Sciences, Engineering and Technology, **2012**, 4(12), 1815-1821.
[9] JesúsOliva; José Ignacio Serrano; María Dolores del Castillo; Ángel Iglesias. Data & Knowledge Engineering, **2011**, 70(4), 390-405.
[10] Montserrat Batet; David Sánchez; Aida Valls. Journal of Biomedical Informatics, **2011**, 44(1), 118-125.
[11] Shaohua Jiang; Jian Zhang; Haiyan Zhang. Journal of Networks, **2013**, 8(5), 2013.
[12] Jin Wang; Engong Chen; Deming Shi. Pattern recognition and artificial intelligence, **2006**, 19(6).
[13] O'Shea.K,. Applied Intelligence, **2012**, 37(4), 558-568.
[14] Lee J; Kim M; Lee Y. Journal of Documentation, **1993**, 49(2), l88-207.