J. Software Engineering & Applications, **2009, 2:55-59**
Published Online April 2009 in SciRes (www.SciRP.org/journal/jsea)

A Semantic Vector Retrieval Model for Desktop Documents

Sheng Li [1]

[1]School of Information, Zhongnan University of Economic and Law, Wuhan, China
Email: kinglisheng@163.com

Received December 3rd, 2008; revised January 29th, 2009; accepted February 18th, 2009.

ABSTRACT

The paper provides a semantic vector retrieval model for desktop documents based on the ontology. Comparing with traditional vector space model, the semantic model using semantic and ontology technology to solve several problems that traditional model could not overcome such as the shortcomings of weight computing based on statistical method, the expression of semantic relations between different keywords, the description of document semantic vectors and the similarity calculating, etc. Finally, the experimental results show that the retrieval ability of our new model has significant improvement both on recall and precision.

***Keywords*:** *Semantic Desktop*, *Information Retrieval*, *Ontology*, *Vector Retrieval Model*

1. Introduction

As an important branch of the semantic Web [1] technology, the semantic desktop indicates the development direction of desktop management technology in the future [2]. In order to implement semantic desktop retrieval, a certain information retrieval model is required, and it is an important research topic of information retrieval. At present, researchers provide a variety of information retrieval model from different angles such as probabilistic retrieval model, fuzzy retrieval model, and vector space retrieval model (VSM) [3]. According to them, the vector space model is the most effective one to express the structure of documents.

The main advantage of traditional vector space model is its simplicity, which could describe unstructured documents with the form of vectors, making it possible to use various mathematic methods to be dealt with. Therefore, we consider using ontology-based semantic information management methods to improve traditional vector space model, creating a semantic vector space model.

2. Traditional Vector Space Model

In the vector space model, the characteristic item t (also known as the index item) is the basic language unit appearing in document d, which could represent some character of the document. The weight of characteristic item is ω_{ik}, which reflects the ability of characteristic item t_k describing document d. The characteristic item frequency tf_{ik} and the inverse document frequency idf_k are used to calculate the value of ω_{ik} with the formula that $\omega_{ik} = tf_{ik} \times idf_k = tf_{ik} \times (\log_2(N/n_k)+1)$, Where tf_{ik} is the frequency of characteristic item t_k in document d_i, and N is the number of documents, n_k is the number of documents that involved the characteristic item t_k. From this formula, we can see that the value of ω_{ik} increases with tf_{ik} and decreases with n_k.

The distance between two document vectors is represented by similarity. The similarity between document d_i and d_j is defined as the cosine of the angle between two vectors:

$$Sim(d_i,d_j) = \cos\theta = \frac{\sum_{k=1}^{m} \omega_{ik} \times \omega_{jk}}{\sqrt{(\sum_{k=1}^{m} \omega_{ik}^2)(\sum_{k=1}^{m} \omega_{jk}^2)}} \quad (1)$$

During the procedure of query matching, the Boolean model could be used to realize the vector conversion of query condition QS.

$$q_j = \begin{cases} 1, & if\ t_j \in QS, \\ 0, & else. \end{cases}$$

The information retrieval algorithm based on the aforementioned basic knowledge is as follows:

1) Creating characteristic item database: Input the characteristic item of documents set, and creating characteristic item database;

2) Creating document information base: Input the content of documents into database, and creating the document information database;

3) Creating document vector database: For each record in document information base, computing its characteristic item weight by formula introduced before, and founding its corresponding document vector;

4) Document query: The user input query condition. Then, acquire eligible document vector by Boolean model, computing the similarity between the query condition and each document by Formula (1);

5) Output the ranking result: According to the similarities computed in step 4), output the query result.

3. The Features of New Model

Though the semantic vector space model draws on some thinking of traditional vector space model, it make some useful improvements based on the specific features of semantic information expression. The main features of semantic vector space model include:

1) The elements and dimension of semantic vector space are different from traditional one. In semantic vector space model, the document characteristic item sequence is not represented by the keywords as usual but the concepts extracted from documents. These concepts contain rich meaning in the ontology. At the same time, for each concept in the concept space, there is a corresponding list to describe. The list represents a vector in the property space. Therefore, each semantic vector in this model is composed of a 2D vector. So, the description capacity of semantic model is better than the traditional one.

2) The method for determining each item's weight is different between semantic vector space model and traditional one. In the semantic model, the weight of an item is related to not only the frequency of a keyword, but also the description of corresponding concept involved in the document. In addition, the TFIDF function in traditional model cannot accurately reflect the distribution of items in the documentation set. In semantic vector space model, the items in different position of a document will be set with different weights. For example, the items appearing in the title of one document will be heavier than the ones appearing in the abstract.

3) The two models use different algorithm to compute the similarity. In the semantic vector space model, the comparability and relativity between two concepts are fully taken into account. For example, in traditional vector space model, the words "People", "Person", and "Human" are totally different concepts, but these words could be conclude as one concept according to corresponding structures or relationships.

4) Besides the differences introduced above, the most important feature of semantic model is the using of ontology as a carrier of information. Comparing with traditional text retrieval methods, the new model involved the semantic information in the ontology.

4. Ontology Creating

Except for the differences introduced in last section, an important character of SVM is the usage of ontology as an information carrier.

The ontology could be seen as a specification of conceptualizations, it defines a group of concepts. Commonly, ontology could be divided into general ontology such as WordNet [4] and domain ontology that describe concepts in some special domain. In this paper, we only focus on ontologies in computer science domain.

4.1 The Relationships in the Ontology

In the ontology, concepts link themselves with other concepts through relationships. In the hierarchical structure graph of ontology, each edge represents a relationship. Three most common relationships are "Is-A", "Part-Of" and "Entity Relationship" [5].

1) Is-A Relationship: It describes the relationship of Generalization. For example, "Entity Extraction" Is-A "Information Extraction";

2) Part-Of Relationship: It describes the containing relationship between concepts. For example, the "CPU" is a Part-Of "Computer";

3) Entity Relationship: It describes the member relationship between a concept and its individual object. For example, "T. Berners-Lee" is an entity of concept "author".

4.2 The Structure in the Ontology

According to the basic principles of ontology and the ACM Topic Hierarchy [6], we create ontology to describe the terms about computer science, called "CmpOnto". Then, the ontology "SwetoDblp_2" is created through the extension of SwetoDblp [7] on the aspect of research field and keywords. The segment of ontology CmpOnto is as follows:

```
<owl:Class
rdf:about="http://www.acm.org/class/1998/acm#H.3">
  <rdfs:label>INFORMATION STORAGE AND
RETRIEVAL</rdfs:label>
  <rdfs:subClassOf
rdf:resource="http://www.acm.org/class/1998/acm#H"/>
</owl:Class>
     ...
<owl:Class
rdf:about="http://www.acm.org/class/1998/acm#H.3.3">
  <rdfs:label>Information Search and Retrieval</rdfs:label>
  <rdfs:subClassOf
rdf:resource="http://www.acm.org/class/1998/acm#H.3"/>
  <owl:disjointWith>
<owl:Class
rdf:ID="http://www.acm.org/class/1998/acm#H.3.1">
      <owl:Class
rdf:ID="http://www.acm.org/class/1998/acm#H.3.2">
   ...
  </owl:disjointWith>
</owl:Class>
```

The segment of ontology SwetoDblp_2 is as follows:

```
<owl:Class
rdf:about="http://lsdis.cs.uga.edu/projects/semdis/opus#Article"
>
```



```
   <rdfs:label>Article</rdfs:label>
   <rdfs:subClassOf
rdf:resource="http://lsdis.cs.uga.edu/projects/semdis/opus#Publ
ication"/>
   <rdfs:comment>An article from a journal or maga-
zine.</rdfs:comment>
   <owl:equivalentClass
rdf:resource="http://knowledgeweb.semanticweb.org/semanticp
ortal/OWL/Documentation_Ontology.owl#Article_in_Journal"
/>
<owl:equivalentClass
rdf:resource="http://sw-portal.deri.org/ontologies/swportal#Arti
cle" />
  <owl:equivalentClass
rdf:resource="http://purl.org/net/nknouf/ns/bibtex#Article" />
</owl:Class>
     ...
<owl:ObjectProperty
rdf:about="http://lsdis.cs.uga.edu/projects/semdis/opus#at_univ
ersity">
  <rdfs:comment>Indicates that a publication originates or is
related to a specific University.</rdfs:comment>
  <rdfs:label>at university</rdfs:label>
  <rdfs:range
rdf:resource="http://lsdis.cs.uga.edu/projects/semdis/opus#Univ
ersity"/>
  <rdfs:domain
rdf:resource="http://lsdis.cs.uga.edu/projects/semdis/opus#Publ
ication"/>
</owl:ObjectProperty>
```

5. Computing the Semantic Similarity

During the procedure of information retrieval based on semantic similarity, the concepts and properties in the vector are processed respectively. Considering the relativity between different conceptual entities and comparable properties, the method for measuring the concept similarity and the property similarity are introduced. Finally, the semantic similarity algorithm was provided.

5.1 The Concept Similarity

Ontology uses hierarchical tree structure to describe the logical relationship between concepts, which is the semantic basis for our retrieval algorithm. Since there is certain relativity between different concepts, we use concept similarity to describe and measure it in order to improve the precision of retrieval. Before computing the concept similarity, we give 3 definitions for different kinds of relationship between concepts as following:

Definition 1: The homology concepts. In the hierarchical tree structure of ontology, concept A and concept B are homology concepts if the node of concept A is the ancestor node of concept B. Call A is the nearest root concept of B, notes as $R(A,B)$; The distance between A and B is $d(A,B) = dep(B) - dep(A)$, where $dep(C)$ is the depth of node C in the hierarchical tree structure.

Definition 2: The non-homologous concepts. In the hierarchical tree structure of ontology, concept A and concept B are non-homology concepts if concept A is neither the ancestor node nor the descendant node of concept B; If R is the nearest ancestor node of both A and B, Call R is the nearest root concept of A and B, notes as $R(A, B)$; The distance between A and B is $d(A,B) = d(A,R) + d(B,R)$

Definition 3: The semantic related concepts. Concept C is the semantic related concept of A and B, if and only if C satisfy the following conditions: If concept A and B are homology concepts, C exists in the sub-trees with root of A but not exists in the sub-trees with root of B; if concept A and B are non-homology concepts, C exists in the sub-trees with root of R, but not exits in the sub-trees with root of A or B.

Figure 1 shows details of the relationships described above. According to these definitions, the structure similarity between concept A and concept B is:

$$Sim(A,B)' = \begin{cases} (1-\dfrac{\alpha}{dep(R(A,B))+1}) \times \dfrac{\beta}{d(A,B)} \times \dfrac{son(B)}{son(A)}, \\ \text{if } d(A,B) \neq 0, \text{and A,B are homology concepts;} \\ (1-\dfrac{\alpha}{dep(R(A,B))+1}) \times \dfrac{\beta}{d(A,B)} \times \dfrac{son(A)+son(B)}{son(R)}, \\ \text{if } d(A,B) \neq 0, \text{and A,B are non-homology concepts;} \\ 1, \text{ if } d(A,B) = 0. \end{cases} \tag{2}$$

where son(C) present the total number of nodes in sub-tree with the root of concept C. The parameter α and βis used to adjust the weight of $dep(R(A,B))$ and $d(A,B)$, whose range is (0, 1), and setting by filed experts.

According to formula given above, the concept similarity decreases with the distance between concepts. At the same time, for two concepts, the deeper the nearest root they have, the more common properties they should have, and the more similar they should be. Further more, the number of nodes in the sub-tree and semantic related concepts are also important factors during the computing of similarity.

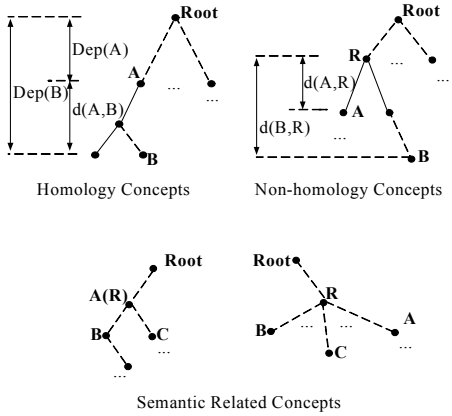

Figure 1. Three patterns of concepts

Finally, the formula defines that the similarity between the same concepts is 1, and the distance between them is 0.

5.2 The Property Similarity

Each concept in the ontology may have several different entities, the main difference among these entities rest with their property values. Further more, different concepts may have same properties. Therefore, not only the concept similarity but also the property similarity should be considered during the computing of similarity between two entities. For the property similarity measuring, we have definition as following:

Definition 4: Suppose I is the entity of concept C, the value of its property $\mathbf{P}_i$ is p_i, i=1,2, . . . , n. Use I=C[P] to present this entity, where P is the property vector $(p_1,p_2,\ldots,p_n)$.

Only the common properties need to process when computing the similarity between property vector $P=(p_1,p_2,\ldots,p_m)$ and $Q=(q_1,q_2,...,q_n)$.

At first, transform the property vectors P and Q into common property vectors $P'=(p'_1,p'_2,....,p'_r)$ and $Q'=(q'_1,q'_2,...,q'_r)$. Then, according to the properties defined in the ontology and the similarity of property value, the property similarity of vector P and Q is given:

$$Sim_p(P,Q)=Sim_p(P',Q')=\sum_{i=1}^{r}\frac{\mu_i+\gamma_i}{2}.Sim_i(p'_i,q'_i) \quad (3)$$

where μ_i and γ_i are weights of property p'_i and q'_i respectively in their property vector, which are preset in the ontology; $Sim_i(p'_i,q'_i)$ is the similarity of property values, which is preset by field expert in the ontology. For example, the similarity between property value "Data mining" and "Information Retrieval" is 0.7, and that between "Data mining" and "Network" is 0.1. The range of $Sim_p(P,Q)$ is [0,1].

5.3 The Semantic Similarity

After computing the concept similarity of semantic vector and the property similarity of conceptual entity, we can get the final semantic similarity of semantic vector.

Suppose $V_1=(A_1[P_1],...,A_m[P_m])$ and $V_2=(B_1[Q_1],...,B_n[Q_n])$ are two semantic vectors. The semantic similarity between V_1 and V_2 is:

$$Sim_V(V_1,V_2)=$$
$$\frac{1}{m}\sum_{i=1}^{m}Max_j\left(\omega\cdot Sim_C(A_i,B_j)+(1-\omega)\cdot Sim_P(P_i,Q_j)\right) \quad (4)$$

where ω is the weight of concept similarity, and its range is [0, 1].

Now, the main retrieval algorithm is as follows:

Begin

1) Initialize the documentation set, then load the user query vector V_1 and deciding its document clustering;

2) Load the semantic index file of documents, initializing the semantic vector V_2;

3) For each vector in the document clustering includes V_1. if current vector has never been processed then continue; else, process the next vector;

4) Compute all the concept similarity between concepts in V_1 and V_2;

5) Compute all the property similarity between concepts in V_1 and V_2;

6) Compute the semantic similarity between V_1 and V_2, insert V_2 into list S with descending order;

7) Output top n items in list S as retrieval results;

End.

6. Experiment and Analysis

In order to verify the effectiveness of our method, we design a prototype system and chose 100 abstracts downloading from DBLP as retrieval target document. In this prototype system, we use ontologies CmpOnto and SwetoDblp_2 introduced in Section 4.

In the experiment, the depth of ontology concept tree is 5, the range of $dep(R(A,B))$ in Formula (2) is [1,5], and the value of $d(A,B)$ is an integer from 1 to 10; both the value of weight α and β is 0.5; the μ_i and γ_i are parameters preset in the ontology, which could be gained by statistical method. The value of ω in Formula (4) will make influence on the retrieval results ranking. In order to choose proper ω, we implement primary experiment for analysis and choosing 0.8 as the optimal value of ω.

The first step of experiment is document pretreatment. Each document is described by a semantic eigenvector V_2 including 1 to 4 conceptual entities. We can find that the average precision of retrieval increase from 60% to 80% according to the increase of concepts in V_2. The corresponding results are shown in Table 1.

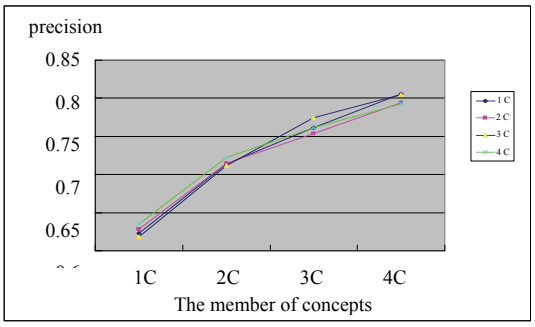

Figure 2. The influence of concepts in V_2 on the precision

Table 1. The influence of concepts in V_2 on the precision

V_1 \ V_2	1C	2C	3C	4C
1 C	0.619	0.711	0.759	0.802
2 C	0.625	0.716	0.752	0.796
3 C	0.630	0.711	0.771	0.803
4 C	0.633	0.724	0.763	0.797

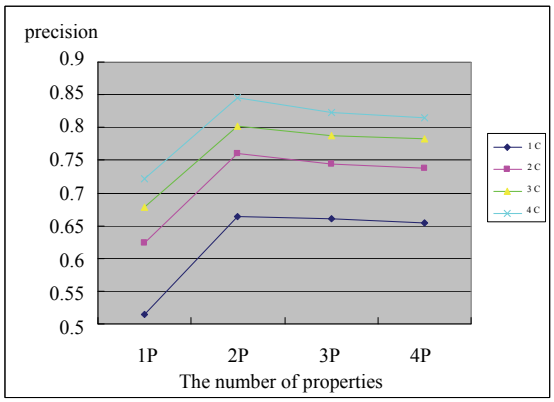

Figure 3. The influence of properties in V_1 on the precision

Table 2. The influence of properties in V_1 on the precision

V_2 \ V_1	1C	2C	3C	4C
1 P	0.513	0.621	0.675	0.721
2 P	0.662	0.764	0.804	0.859
3 P	0.649	0.741	0.785	0.821
4 P	0.637	0.739	0.781	0.811

Table 3. The comparing of different retrieval models

Precision / Documents	Keywords Retrieval	Semantic Retrieval
5	74.2%	88.3%
10	66.5%	82.5%
15	58.4%	75.5%
20	50.3%	71.2%
25	43.9%	65.8%
30	37.7%	58.5%
35	34.3%	50.4%
40	27.4%	47.2%
45	21.6%	42.9%
50	19.4%	36.3%
Average	43.37%	61.86%

Figure 2 could reflect the relationship between the number of concepts in V_2 and the precision of query more directly.

Further more, statistical results show that the number of properties in the conceptual entity could also make influence on the precision. When the number of properties is 2, the effects go best. If a concept has too many properties, some proper target will be missed because of so many restrictive conditions. The corresponding results are shown in Table 2.

The Figure 3 is corresponding to Table 2.

In addition, we compare our new model with traditional VSM model based on keywords. The number of documents and the precision of retrieval are shown in Table 3. The average precision of semantic retrieval is 61.86%, but only 43.37% by traditional method in the same documentation set. According to the experimental data and analysis above, we know that the ontology could play a positive role in upgrading the precision of retrieval.

7. Conclusions

This paper provides a semantic retrieval model based on the ontology for desktop documents. Comparing with traditional vector space model, the new model using semantic and ontology technology to solve a series of problems that traditional model could not overcome. The experimental results prove the effectiveness of this new model.

In addition, the individual analyses for retrieval results tell us that there is little distinction in result ranking by different retrieval methods. The main reason for precision upgrading is that the semantic retrieval method could reduce the similarity of incorrect results, so that the correct result could be ranked in the front position. Therefore, how to re-rank and optimize the retrieval results is an important task, and it is our main item in the next stage.

REFERENCES

[1] B. Lee, Hendler, and Lassila, "The semantic web," Scientific American, Vol. 34, pp. 34−43, 2001.

[2] S. Decker and M. Frank, "The social semantic desktop," WWW 2004 Workshop Application Design, Development and Implementation Issues in the Semantic Web, 2004.

[3] I. R. Silva, J. N. Souza, and K. S. Santos, "Dependence among terms in vector space model," Database Engineering and Applications Symposium, pp. 97−102, 2004.

[4] G. A. Millet, "Wordnet: An electronic lexical database," Communications of the ACM, 38(11): pp. 39−41, 1995.

[5] G. Asian and D. McLeod, "Semantic heterogeneity resolution in federated database by metadata implan- tation and stepwise evolution," The VLDB Journal, the International Journal on Very Large Databases, Vol. 18, pp. 22−31, 1999.

[6] ACM Topic: http://www.acm.org/class/.

[7] B. Aleman-Meza, F. Hakimpour, I. B. Arpinar, and A. P. Sheth, "SwetoDblp ontology of Computer Science publications," Web Semantics: Science, Services and Agents on the World, pp. 151−155, 2007.

