Research on Information Retrieval Based on Semantic Similarity Related Technology

Yangxin YU[1], Yizhou ZHANG[2]

[1] Faculty of Computer Engineering, Huaiyin Institute of Technology
Huai'an, 223003, China

[2] Party School of Chinese Communist Party Huai'an City Committee
Huai'an, 223003, China

Abstract

In this paper, an information retrieval method based on knowledge reasoning that tightly integrates description logic reasoning and traditional information retrieval technique is proposed. The method expresses the user's search intention by description logic to infer the user's search object. Further, fuzzy describing logic is introduced to confirm the relations between Web pages and user's search requirement. In order to calculate the semantic similarity, we take property values and multi-inheritance of entities into consideration, and optimize the computing process based on the tree structure of inheritance relationship.When the instance multiple inheritances are relatively complex, the accuracy rate is more pronounced than existing methods.The experimental results show that the scheme proposed in this paper can calculate semantic similarity more accurately even in ontology-based knowledge base.

Keywords: *Knowledge Reasoning, Information Retrieval, Fuzzy Description, Tree-structured, Multi-inheritance,Semantic Similarity.*

1. Introduction

With the rapid development of Web technology, the Internet has become a huge information database of globalization, people are becoming more and more dependent of search engine when they get information from the Internet. Based on keyword matching and linking relationships of traditional search engines, a large number of irrelevant results with user queries are often returned for information retrieval, because its content can not really understand the user's requirements intention. It is hoped that a breakthrough in information retrieval technology appears, the information retrieval technology should support the more powerful information retrieval capabilities with the understanding of semantics, automatic expansion, association ability, and can provide personalized services with users. From the application sense, semantic retrieval need transform the user's query into a semantic concept by semantic understanding and computing, which retrieve relevant to this concept, the information the user really wants and overcome the limitations of traditional information retrieval techniques. In the current Internet environment, semantic retrieval need to solve two technical problems: the semantics of user needs and Web content, which gives the exact content of the user needs and machine-understandable meaning. In this paper, the major research work is the introduction of fuzzy description logic to information retrieval, Web content of the user needs and issues of the semantics, an information retrieval methods of knowledge reasoning is proposed, named as *IRKR*.

The traditional information retrieval techniques determine the matching degree between Web pages and user query requirements by calculating the space vector similarity between Web pages and query keywords[1].As ontology knowledge base is commonly used to express the concept of the user's query requirements in semantic search methods. People need to analyze the semantic similarity between concepts and determine the relevance between concept and user requirements. During the research of semantic information retrieval, the calculation of semantic similarity can be divided into three categories:Path length,Information theory and Concept characteristics.

1.1 Path length method

This type method uses the path length between two concepts of ontology knowledge base to infer the semantic similarity between instances. The shorter the path length is, the greater the semantic similarity between concepts is. Reference[2] extended the original method to give the different weights in the edge path in order to improve the accuracy of semantic similarity calculation.

1.2 Information theory method

This type method infers the semantic similarity degree between the concepts according to the relationship of the

concept information content. Reference[3] uses hierarchical structure to express the similarity degree of concept t_1 and t_2 with the maximum of concept information content. In which, the information content of the concept t=-logPr[t], Pr[t] is the probability belonging to the concept t of any instance in the knowledge base. Reference [4] is extended based on reference[3], and proposed the similarity degree formula $\sigma(t_1,t_2)$ of concept t_1 and t_2. $\sigma(t_1,t_2)$ is calculated as Eq.(1):

$$\sigma(t_1,t_2) = \frac{2 \times \log \Pr[t_0(t_1,t_2)]}{\log \Pr[t_1] + \log \Pr[t_2]} \quad (1)$$

Where, $t_0(t_1,t_2)$ is the deepest common ancestors of concept t_1 and t_2 in the hierarchical structure.The advantage of semantic similarity based on information theory method lies in its theory as a foundation of information theory, but this method can not be applied to the situation there is a clear relationship between data sparse in ontology knowledge base and information elements.

1.3 Concept characteristics method

This type method determines the similarity between concepts by comparing concept with the specific property value. Such as reference[10] uses synonym sets, distinguishing features, semantic neighborhoods and other kinds of property conditions as a basis for comparison in the analysis case.

All the above methods are the three different perspectives to analyze the similar situation of the concept.In the actual calculation of semantic similarity, it can be necessary to make the above three categories integrate.

In the analysis of the traditional semantic similarity method, we found that these methods were ignoring an important fact. Semantic information retrieval in the hierarchy of organizations using ontology knowledge base is often more complex example of inheritance, that is an instances may inherit from multiple classes. But existing semantic similarity calculation method are not considered instances of multiple inheritance impacting on the similarity calculation, the relationship analysis between the level of attributes and their values on the instance is not entirely, and thus seriously affects the accuracy of calculation of semantic similarity. For the shortage of the existing similarity measure. This paper first discusses the impact of semantic similarity analysis on the instances multiple inheritance and the instances properties hierarchical, and presents a comprehensive similarity calculation method with multiple inheritance and property value factors in the hierarchy, and the validity of the method is verified by experiment.

2. Fuzzy Description Logic

Description logic(*DL*) is an object-based knowledge representation theory, also called the concept language or terms logic[4]. It is a decidable subset of first-order logic, a suitable semantic definition, and with strong expressive power. A description logic system contains four basic components: the construction set of concepts and relationships of concepts, the terminology set of Tbox assertion, individual set of Abox assertions and the reasoning mechanism of Abox and Tbox. Expression of a description logic system capacity and reasoning ability depends on the choice of the above-mentioned several factors and different assumptions.

Description logic knowledge base Σ contains two components: Tbox and Abox, as $\Sigma=(T, A)$, where "T" represents Tbox and "A" represents Abox. Tbox is axioms set of the description field structure and Abox is axioms set of the description of the specific individuals facts. Generally, description logic constructs complex concepts and relationships in the simple concepts and relations according to the provided structure operator. Usually description logic contains the following operators at least: $\wedge$, $\vee$, $\neg$, $\exists$ and $\forall$. This basic description logic is called the *ALC*[5]. On the basis of the *ALC*, different structure operator are added to the description logical to constitute different expression.

Fuzzy description logic describes the uncertainty of description logic. The traditional description logic is based on binary judgments, if an individual "a" is a member of the class "A", the answer is only "*YES*" or "*NO*". In the fuzzy description logic, the subordination of the individual "a" is uncertainty. If the degree of membership of individual "a" is *0.8*, the individual is likely to be a member of the class "A".

3. User's Search Intention

Query requirements is a description of users concerning object or object property. The information of query requirements can be divided into two categories according to different functions:

Object constraint, it illustrates that query object need meet the conditions and constraints;

Attribute constraint, it illustrates that users concerned about what information of the query object.

IRKR describedes the object constraint of query requirements with *ALC* concept, set description attribute of *ALC* role. The concept of object constraint in query requirements is represented in the form of object constraint (*OC*).

$OC=D_1 \vee D_2 \vee \ldots \vee D_n$

$D_i=A_{i1} \wedge A_{i2} \wedge \ldots \wedge A_{im}$ $(1 \le i \le n)$

Where D_i is called sub-object constraint, and A_{ij} ($1 \le j \le m$) is called as atomic object constraint. Atomic object constraint is the basic constituent unit of the object constraint.

Attribute constraint of query requirements is represented by the set $P=\{p_1, p_2, ..., p_k\}$, where p_i is called as atomic property constraint, corresponding to an property of query object.

Supposed, OC of query requirements "s", $OC=D_1 \vee D_2 \vee ... \vee D_n$, $n>0$, property constraint $P=\{p_1, p_2, ..., p_k\}$, $k \ge 0$. When $k>0$, the combination of D_i ($1 \le i \le n$) and p_j ($1 \le j \le k$) is a sub-query needs "s". When $k=0$, D_i is a sub- query needs "s". Web pages is related to any sub-query requirements "s", that is related to user's query requirements.

When the role of atomic property constraint or object constraint is beyond the scope of knowledge base, *IRKR* automatically addes to the knowledge base in the property form, called as the expanded role. When the concept of object constraint is beyond the scope of knowledge base, *IRKR* automatically addes to the knowledge base in the concept form, called as the expanded concept. The expanded role and the expanded concept are not related to any example of knowledge base.

Supposed, the query requirements "s" is user evaluation and dealers of "*Nokia 1110*" . Where "*Nokia 1110*" illustrates the object of user queries, called as object constraint information. User evaluation and dealers illustrates the property of query object concerned by users, called as property constraint information. Object Constraint of "s" is $OC=D_1=Cell\ Phone \wedge \exists manufacturer.Nokia \wedge Name.1110$, property constraint of "s" is $P=\{Retailers, User\ comment\}$. There are two sub-query requirements:

$s_1=\{Cell\ Phone \wedge \exists manufacturer.Nokia \wedge Name.1110, Retailers\}$

$s_2=\{Cell\ Phone \wedge \exists manufacturer.Nokia \wedge Name.1110, User\ comment\}$

Where "*User comment*" is beyond the scope of knowledge base, called as the expanded role.

4.Web Associated with User Queries

4.1 Semantic Association.

Semantic association describes the close degree of semantic relationship between entities (including class, property, instance). In the ontology knowledge base, if there is one or more properties the sequences between two entities, there is semantic association[6]. Considering the semantic relationship, the relevance between Web pages and instances can more effectively determine. There is a strong semantic association between "*Nokia*", "*GSM*", "*Cell Phone*" and "*1110*" in Fig.1. When we determine the relevance between Web pages and the instance "*1110*", we can not accurately determine the corresponding because of the ambiguity of keywords (the entity name as the entity key). In this case, the keyword related to other entities appearing in Web pages with the instance "*1110*" , such as "*Nokia*", "*GSM*", "*Cell Phone*", will increase the possibility of the Web pages with the instance "*1110*" and directly represents with *R (a, b)*, the semantic association value of instance "*a*", "*b*".

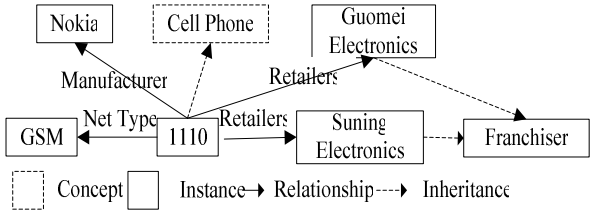

Fig. 1 Knowledge base clips

4.2 Correlation of Web pages and Instance

Semantic relationship between the synthesis instance can more accurately analyze the relevance of Web pages and instance. To reduce the computational complexity, *IRKR* only considers the entity γ (γ specified by the user), whose semantic association degree is larger than given instance "*a*". Supposed the entity set is $T(a)=\{a_1, a_2, ..., a_n\}$, and $a \in T(a)$. The relevance of Web "*p*" and instance "*a*" is as Eq.(2):

$$relationDegree(a,p)=\begin{cases}1, \sum_{i=1}^{n} relation(p,a_i)*\frac{R(a,a_i)}{R(a,a)}>1 \\ \sum_{i=1}^{n} relation(p,a_i)*\frac{R(a,a_i)}{R(a,a)}, else\end{cases} \quad (2)$$

$relation(p, a_i)$ is keyword vector similarity between entity "*a*" and Web "*p*" (When a_i is an instance, it's text is illustrated the basic keywords, and taken three classes and property description are extracted as the keyword expansion. When a_i is a class, it's text is directly illustrated the keywords) as Eq.(3):

$$relation\ (p,a_i)=\frac{\sum_{j=1}^{m} td_j \times \mu tc_{ij}}{\sqrt{\sum_{j=1}^{m} td_j^2 * \sum_{j=1}^{m} \mu tc_{ij}^2}}, 1 \le i \le 1 \quad (3)$$

Where m is the total number of keywords for the Web page text and a_i entities, td_j and tc_{ij} is keyword value, respectively. μ is weight of the basic keywords and extended keywords. When tc_{ij} is the basic keywords, $\mu=2$. When tc_{ij} is the extended keywords, $\mu=1$. The relevance of Web "*p*" and instance "*a*" is keyword similarity sum of Web "*p*" associated with instance "*a*" in set *T(a),* and maximum value is *1*. The semantics between a_i and

instance "*a*" is stronger, the correlation degree of *relation(p, a_i)* on Web "*p*" and instance "*a*" is greater.

4.3 Membership of Web pages and Concept.

By analyzing the correlation of Web and instance, *IRKR* can derive the membership of Web pages and concept in knowledge base.

Requirements of sub-query corresponding to the concept: *SC*.

When *SC* is atomic concept, supposed $B=\{b_1, b_2, \ldots, b_x\} \subset \triangle^I$ ($\triangle^I$ is the interpretation field), $SC^I(b_i)=1(1\le i \le x)$, the membership of Web "*p*" and concept *SC* is as Eq.(4):

$$SC(p)=\begin{cases} 1, \sum_{i=1}^{x} relationDe\ \ gree(b_i, p) > 1 \\ \sum_{i=1}^{x} relationDe\ \ gree(b_i, p), \quad else \end{cases} \tag{4}$$

According to characteristics of fuzzy description logic, if *SC* is compound concept, there exists $SC_i^I(p)=(DC_i \wedge PC_i)^I(p)= min\ \{DC_i^I(p), PC_i^I(p)\}$.

Sub-object constraint corresponding to the concept: *DC*.

When $\exists\, a_i \in \triangle^I$ and $DC^I(a_i)=1$, supposed $A=\{a_1, a_2, \ldots, a_t\} \subset \triangle^I$ and $DC^I(a_i)=1$, the membership of Web "*p*" and concept *DC* is as Eq.(5):

$$DC^I(p)=\begin{cases} 1, \sum_{i=1}^{n} relationDe\ \ gree(a_i, p) > 1 \\ \sum_{i=1}^{n} relationDe\ \ gree(a_i, p), \quad else \end{cases} \tag{5}$$

When $\forall a_i \in \triangle^I$ and $DC^I(a_i)=1$, supposed D_i is sub-object constraint corresponds to *DC*, taken text description of classes and properties in D_i as query keywords of concept *DC*. $DC^I(P)$ is the keyword term frequency vector similarity of keyword query of Web text associated with *DC*, calculated as Eq.(5).

Atomic property constraint corresponding to the concept: *PC*.

Supposed p_i is the atomic property constraint corresponds to the concept *PC*, the text message of p_i is taken as query keyword of concept *PC*. $PC^I(P)$ is the keyword term frequency vector similarity of keyword query of Web text associated with *PC*, calculated as Eq.(5).

Needs of the user query corresponding to the concept: *S*

According to characteristics of fuzzy description logic, supposed $S = SC_1 \vee SC_2 \vee \ldots \vee SC_n$, there exists $SC_n)^I(p)= \max\ \{ SC_1^I(p), SC_2^I(p), \ldots, SC_n^I(p)\}$.

5. Multiple Inheritance Analysis

In real life, it is very common for a thing with multiple identities.For example , *Lee* may have multiple identities as math professor,*WuShu* hobbies, and party member.In the ontology knowledge base, this phenomenon is an instance belonging to multiple classes.People ignore the case of multiple inheritance in previous semantic similarity calculation when they consider inheritance and simply select a map from multiple identities , lead to inaccurate similarity calculation[7]. In Fig.2, instance *A* and instance *B* are *computer professor* and *Ping-pong enthusiasts* , instance *C* is only *computer professor*. If we only consider a inheritance in semantic similarity calculation, such as we only consider instance *A* and instance *B* as *computer professor*, then *sim(A,B)=sim(A,C)* (*sim(M,N)* represents the similarity between instance *M* and instance *N*). But instance *A* and *B* are also *Ping-pong enthusiasts* , *sim(A,B)>sim(A,C)* actually.

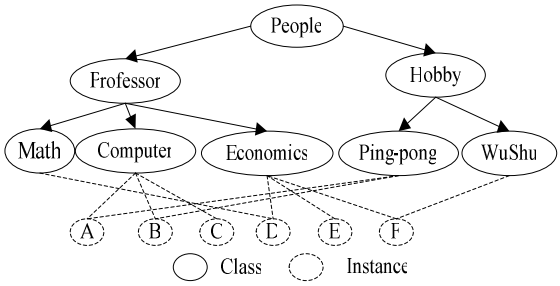

Fig. 2 Entity multiple mapping.

The most direct solution is to cumulative the generated similar of instance with multiple inheritance.However, a simple superposition may be an error to increase the similar of some examples relations. If we consider the similarity between instance *D* , instance *E* and instance *C*, and cumulative the generated similar of instance with multiple inheritance , *sim(C,D)=2*sim(C,E)*.Whether we consider instance *D* as a math professor or economics professor, the professor identity is similar to instance *C* and *E* , thus *sim(C,D)=sim(C,E)*. If we consider the similarity between instance *F* and *C* , instance *F* with economics professor is similar to instance *C* as the professor , and instance *F* with *WuShu* enthusiast is similar to instance *C* as the people . Because the professor is also sub-category of people , the similarity between instance *F* and *C* can not be added , thus we should not consider the similarity again in calculating the similarity.

6. Property Level Analysis

The property of ontology knowledge base has the same level, property value because of their inheritance lies in the level structure of the ontology knowledge base, and the properties of this level will affect the calculation of the instance similarity.

In Fig.3 , the property of instances *A, B, C, D* follows as:

A:has_CD(c_1)
B:has_DVD(b_1)
C:has_CD(c_2)

D:has_CD(d_1)

Analyzing the similarity between *B, C, D* and *A*. *C, D* and *A* have the same properties, but property values are different, the property and the property values of *B* and *A* are different. Based on public information by the traditional method of determining similarity, *C, D* and *A* in addition to the same man, the only similarity is that there *has_CD* properties; *B* and *A* in addition to the same man, but does not have other similarities.

Observe the relationship between the entities described in Fig.3 and find that *has_CD* and *has_DVD* are the sub-properties of *has_item* that people belong to the relationship between culture and entertainment products. *c1, c2*, *d1* belong to rock, rock is the sub-class of culture and entertainment products, namely *c1, c2, d1* are cultural and entertainment products. *b1* is a documentary and is also cultural and entertainment product. Based on this, you can infer that *C*, *D* and *A* have the rock CD, *B* and *A* have the entertainment product. This is the similarity between instances. Further analysis, *C* and *A* are the *Beatles_CD*, the similarity is maximum; the *Beatles_cd* of *D* and the *Beatles_cd* of *A* belong to the same rock, and is more similarity than *B* and *A*. That is *sim (C, A)> sim (D, A)> sim (B, A)*.

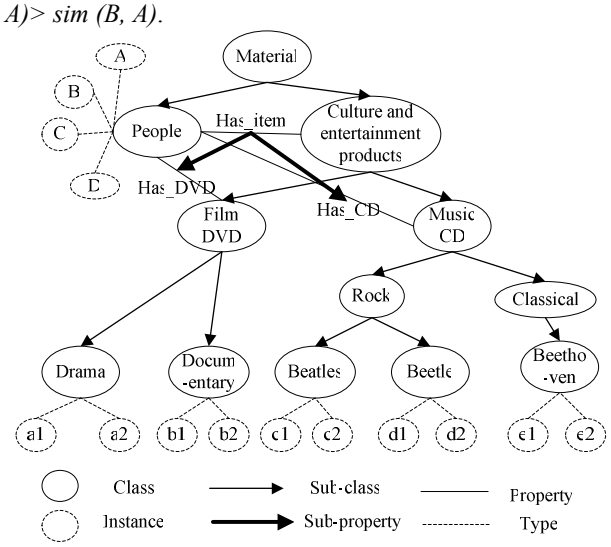

Fig. 3 Instance-level relations.

7. Semantic Similarity Calculation Method

7.1 Instance Inheritance Relation Similarity.

Query requirement is a description of users concerning object or object property . The information of query requirements can be divided into two categories according to different functions : Instance inheritance relation similarity is determined by the instance location in the ontology knowledge base and the similarity of instance inheritance relation[8].

For entity L_1 and entity L_2 , $L_1 \neq L_2$, directly inherited class of entity L_1 is L_{11} ,L_{12} , L_{13} , ... , L_{1m} , and directly inherited class of entity L_2 is L_{21} , L_{22} , L_{23} , ..., L_{2m} . If we ignore the other inheritance relation , only consider entity L_1 and entity L_2 as the class of L_{1i} , L_{2j} ($1 \le i \le m$, $1 \le j \le n$) respectively , the similarity is as Eq.(6) :

$$L_{1i} \bullet L_{2j} = \frac{2 * depth\ (LCA\ (L_{1i}, L_{2j}))}{depth\ (L_{1i}) + depth\ (L_{2j})} * \alpha_{1i} * \alpha_{2j} \qquad (6)$$

depth(L) is the depth of class *L* in the ontology knowledge base . $LCA(L_{li},L_{2j})$ is the largest common ancestor of the class L_{li} , L_{2j} . $\alpha_i = -logPr[L_i]$ is the information content of the class L_i , $Pr[L_i]$ is the probability of instance belonging to the class L_i . Eq. (6) illustrates that the greater the depth of the common ancestor for class L_{li},L_{2j} is , the greater the similarithy is . When $L_{li}=L_{2j}=LCA(L_{li},L_{2j})$, the largest similarity is $\alpha_{1i} * \alpha_{2j}$. Comprehensive other inheritance, instance inheritance relation similarity of l_1 , l_2 is defined as as Eq. (7):

$$sim\_inherit\_relation(l_1, l_2) = \sum_{i=1,j=1}^{i=m,j=n} (L_{1i}, L_{2j}) * \beta \qquad (7)$$

From the above analysis, when the $LCA(L_{li},L_{2j})$ and its subclasses first appeared in the calculation, $\beta=1$. Otherwise , $\beta=0$.

Because of the β impact, selection different combination order of instance inheritance relation in Eq.(7) will get the different similarity of instance inheritance relation. The maximum of $sim\_inherit\_relation(l_1,l_2)$ is the optimize similarity of instance inheritance relation.The *sim_inherit_relation* represents the optimize similarity of instance inheritance relation in the later writings.

7.2 Instance Property Similarity.

Instance property similarity is determined by the relations of instance property.

For instance l_1 and instance l_2 , $l_1 \neq l_2$, $A=\{\mu_{a1}, \ ... \ ,\mu_{an}\}$ and $B=\{\mu_{b1}, \ ...,\mu_{bm}\}$ are direct property set of instance l_1 and instance l_2 respectively (instance property is excluded due to property inheritance). The property similarity between μ_{ai} and μ_{bj} is Eq. (8):

$$sim\_between\_property(\mu_{ai}, \mu_{bj}) = \frac{2 * depth\ (LCA\ (\mu_{ai}, \mu_{bj}))}{depth\ (\mu_{ai}) + depth\ (\mu_{bj})} * \alpha_{ai} * \alpha_{bj} \qquad (8)$$

$depth(\mu)$ is the depth of property μ in the properties hierarchy. $LCA(\mu_a,\mu_b)$ is the largest common ancestor of the property μ_a and μ_b. $\alpha_i = -logPr[\mu_i]$ is the information content of the property μ_l , $Pr[\mu_i]$ is the probability of instance relation belonging to μ_l . Eq. (8) illustrates that the greater the depth of the common ancestor for property μ_a

and μ_b is, the greater the similarithy is. When $\mu_a=\mu_b=LCA(\mu_a,\mu_b)$, the largest similarity is $\alpha_a*\alpha_b$. Instance property similarity of l_1 and l_2 is defined as Eq. (9):

$$sim\_property(l_1,l_2)=\sum_{i=1,j=1}^{i=n,j=m} sim\_between\_property(\mu_{ai},\mu_{bj})\times\eta^{count-1} \quad (9)$$

$\eta\in[0,1]$, *count* represents the number of occurrences for $LCA(\mu_{ai},\mu_{bj})$ and sub-property, and indicates that the similarity of the number of occurrences n is less than the similarity of the number of occurrences $n-1$ for property.

Because of the η impact , selection different combination order of property inheritance relation will get the different similarity of instance property . The maximum of $sim\_property(l_1,l_2)$ is the optimize similarity of instance property. The *sim_property* represents the optimize similarity of instance property in the later writings .

7.3 Instance Property Value Similarity

Instance property value similarity is determined by instance property values relations.

For instance l_1 and l_2 , $A=\{\mu_{11},...,\mu_{1n}\}$ and $B=\{\mu_{21},...,\mu_{2m}\}$ are property set of instance l_1 and instance l_2 , and $C=\{r_{11},...,r_{1n}\}$ and $D=\{r_{21},...,r_{2m}\}$ are the corresponding property values set respectively .

The calculation of property values for r_{1i} and r_{2j} (*Sim_between_property*_result) is similar to instance inheritance relation of r_{1i} and r_{2j} .Just *sim_inherit_relation* in Eq.(7) becomes as Eq.(10):

$$L_{1i}\bullet L_{2j}=\begin{cases}\dfrac{2*depth(LCA(L_{1i},L_{2j}))}{depth(L_{1i})+depth(L_{2j})+2}*\alpha_{1i}*\alpha_{2j}, & r_{1i}\neq r_{2j}\\ \alpha_{1i}*\alpha_{2j}, & r_{1i}=r_{2j}\end{cases} \quad (10)$$

For instance l_1 and instance l_2 , property value similarity is as Eq.(11):

$$sim\_property\_value(l_1,l_2)=\sum_{i=1,j=1}^{i=n,j=m}(sim\_between\_property(\mu_{1i},\mu_{2j})\times sim\_between\_property\_rusult(r_{1i},r_{2j})) \quad (11)$$

7.4 Instances Semantic Similarity

Comprehensive the similarity of instance inheritance relation , instance property and property value , semantic similarity of instance l_1 and l_2 is defined as Eq.(12):

$$\begin{aligned} sim(1_1,1_2)&=\theta*sim\_inherit\_relation(1_1,1_2)+\lambda*sim\_property(1_1,1_2)+\gamma*sim\_property\_value(1_1,1_2) && I_1\neq I_2\\ sim(1_1,1_2)&=Max\_Similarity && I_1=I_2 \end{aligned} \quad (12)$$

Where, *Max_Similarity* is the maximum value of the instance similarity . $0\le\theta$, λ and $\gamma\le1$, $\theta+\lambda+\gamma=1$. Specific value is determined by the application .

8. Algorithm Optimization.

For instance l_1 and instance l_2, $l_1\neq l_2$, l_1 directly inherits from m classes and l_2 directly inherits from n classes . When we calculate the similarity of instance inheritance relation , if we try to find the maximum of all of the selection orders of inheritance relation, then the total calculation is $(m\times n)!$ kinds of possible[9].

The inheritance similarity derived tree of instance l_1 and instance l_2 exists in the ontology inheritance relation structure, directly inherited class of l_1 and l_2 is regarded as the branch of a leaf node and remove the other branch so as to get the inheritance relation tree[10]. Fig.4 is an inheritance similarity derived tree of instance D and F in Fig.4.

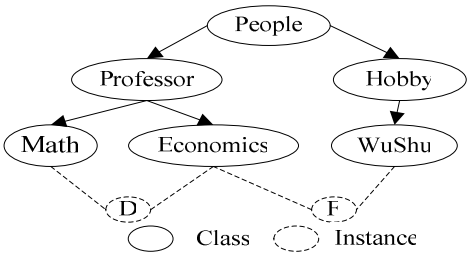

Fig. 4 Instance inheritance similarity derived tree.

The following pseudo code shows the algorithm to ensure the less computational efforts and identify the most optimal similarity of the corresponding combination order of the instance inheritance . Algorithm is as follows :

Algorithm input: instance l_1 and instance l_2 ;

Algorithm output: double instance inheritance relation similarity .

Procedure *similarityByInheritRelation*(1_1,1_2){
1: Vector *vec1 ;* Vector *vec2* ; ArrayList *List* ;
2: Class *classroot* ;
3: Find the direct inheritance classs of instance l_1 and l_2 respectively, and save to the vector *vec1,vec2* ;
4: Generate the inheritance similarity derived tree of instance l_1 and l_2 , and *Root node* is *classRoot* ;
5: for(int $i=0$; $i<$ the members number of *vec1* ; i++)
6: for(int $j=0$; $j<$ the members number of *vec2;j*++)
{
7: Construct an new *InheritSimilarity* class;
8: Calculate the similarity of *vec1.elementAt(i)* and *vec2.elementAt(j)* in Equation (1) and save to *InheritSimilarity.similarity* ;
9: Save the the largest common ancestor of the *vec1.elementAt(i)* and *vec2.elementAt(j)* depth to *InheritSimilarity.commonFather* ;
10: Save *InheritSimilarity* to *list* ;
}
return *similarityInDeduceTree* (*classroot,list*) ;
}

Algorithm input: inheritance similarity derived subtree class of similarity inherited class and class similarity series .

Algorithm output: double instance 1_1 and 1_2 inheritance relation similarity in the derived subtree

```
Procedure similarityInDeduceTree (Class root,
  ArrayList list) {
11: double similaritySubtree=0 ;
12: double similarityRootClass=0 ;
13: SimilarityRootClass is the maximum similarity of
    the deepest common ancestor class as root ;
14: Regard direct subclasses of the root as derived
    subtree of tree root , use similarityInDeduceTree
    to calculate the similarity , and save cumulative
    similarity to similaritySubtree ;
15: if(similarityRootClass>similarityInDeduceTree)
16:    return similarityRootClass ;
17: else
18:    return similaritylnDeduceTree ;
}
19: class InheritSimilarity{
Class commonFather ;
double similarity ;
    }
```

When the algorithm takes a combination of inheritance relation as $\beta=1$, it will be guaranteed that the similarity is greater than the maximum similarity that can be achieved as $\beta=0$, and vice versa . The resulting similarity is the optimization instance inheritance relation similarity of the instance of instance 1_1 and 1_2 , and the calculation is only derived once in traversing the inheritance similarity derived tree.

9. Experimental Results and Analysis

9.1 Prototype Design.

There are differences between the examples and concepts in *ALC*, and the relationship between Web pages and query requirement exists the ambiguity, that can not accurately determine whether a page is relevant to the query requirement or not, it only analyzes the possibilities of Web pages relevant to the query requirement. The fuzzy description logic *FALC*[11] is used in *IRKR*, and describes the correlation of this ambiguity between Web pages and query requirement: The user's query requirement is considered as the concept *FALC*, and Web pages is considered as the example *FALC*, thus the related judgment problem to Web pages and user requirement is turned into the membership degree calculation of concept between Web instance and query concept. To verify the validity of the method *IRKR*, we designed a prototype system corresponding *SSKR*(Search System Based on Knowledge Reasoning), this system structure is shown as Fig.5.

Search Service Portal receives query requirements *Q*, and the list of Web pages in descending order is returned according to the size of the concept membership. *ALC* Engine is a reasoning engine of standard description logic and is implemented based on Jena[12] API. *IR* engine accesses traditional search engines by calling the *Google* Web APIs, and obtaines the relevant Web pages. Relevance assertions module calculates the membership between Web pages and fuzzy concept. Knowledge base can be any *OWL-DL* knowledge base, and the used knowledge base in this paper is to extend the open ontology knowledge base on the basis of *SWETO*[13]. *FALC KB* is a fuzzy extension of knowledge base, and *FALC* engine is a fuzzy description logic reasoning engine.

The process of *SSKR* query is as follows:

the user submits a query requirement *Q*;

ALC engine generates a query keywords *Q'* corresponding to *Q* according to ALC reasoning;

IR engine recalls the Web pages related to *Q'*;

Relevance assertions module calculates the membership between Web pages and fuzzy concept, and generates the fuzzy description logic knowledge base *FALC KB*;

FALC engine calculates the membership between Web pages and query requirements, and sorts the Web pages.

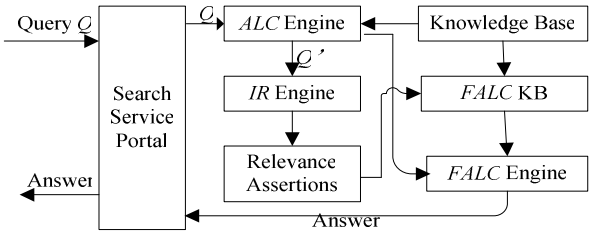

Fig. 5 SSKR System structure.

9.2 Performance Results Analysis

Instance of the user query (concept) is changed into the form of keywords using *WSTT*[14], and is sorted on the recall Web pages by *IR* query method. In this section, we verifies the quality of generated query keywords and the effectiveness of judgments related to methods using *SSKR* by comparing with *WSTT*. We recruites *50* volunteers in Faculty of Computer Engineering, Huaiyin Institute of Technology, each volunteer puts forward *20* query requirements (To meet the WSTT implementation conditions, gained examples set by query requirements reasoning on the *ALC* is not empty) , and are queried by *WSTT* and *SSKR* respectively (*WSTT* query is carried out after the *ALC* reasoning on query requirements) .

The results are shown in Fig.6, *WSTT1* is the generated keywords using *WSTT*, and is reordered on recall Web

pages with *WSTT* method. *SSKR1* is the generated keywords using *SSKR*, and is reordered on recall Web pages with *SSKR* method. The query accuracy of top *20* results of *SSKR1* and *WSTT1* is *46.2%* and *41.1%*, respectively. Recall of *SSKR1* better than *WSTT1* indicates that the generated query keywords by *SSKR* can more accurately convey the user's query requirements and eliminate the ambiguity of keywords, and the *SSKR* model based on semantic association and fuzzy description logic can more effectively determine the relevance of Web pages and query requirements comparing with the *WSTT*.

Comparing with other similarity calculation method in this section, the proposed method in this paper are relatively rich in the effectiveness calculation. instances between multiple inheritance and property relation in the ontology knowledge base .

51 different attractions are selected from '*Destination Guide*' of the *e Dragon* network (*www.elong.com*) , and establishs ontology base about attractions location , climate, consumption levels and attractions feature . *198* instances and *1149* property relations constitute the ontology base .

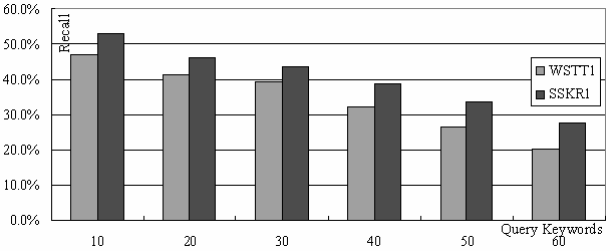

Fig. 6 Query performance analysis.

5 attractions such as $a_1 ,...,a_5$ were randomly selected from *51* attractions . For every attraction a_i $(1 \le i \le 5)$, we can manually get the top *5* similarity to a_i from the remaining attractions,denoted as $B=\{b_{i1},...,b_{i5}\}$. In addition to $\{a_i ,b_{il} ,...,b_{i5}\}$, *N* attractions are selected from the remaining attractions,denoted as $C=\{c_{i1} ,...,c_{iN}\}$, $1 \le N \le 45$. We calculate the $\{b_{i1} ,...,b_{i5} ,c_{i1} ,...,c_{iN} \}$ similarity to a_i using the proposed method in this paper , the *GCSM*[15] and the *ADSS*[16] method respectively so that we can get the percentage of the top *5* similarity attraction belonging to $\{b_{i1} ,...,b_{i5}\}$. The higher the percentage is , the higher the method accuracy is . Experiments were carried out on $a_1 ,...,a_5$ respectively, and the mean percentage was showed as Fig.7.

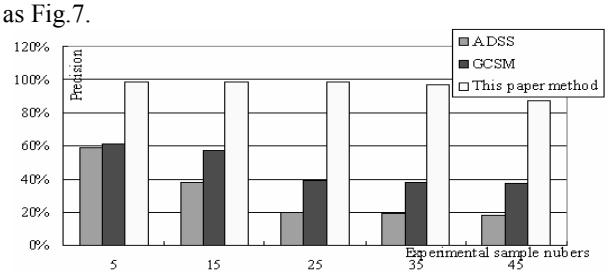

Fig.7 Attractions similarity judgement results

In Fig.7, $N=45$ (*N* is the number of experimental samples) , accuracy of this paper method is *84%* , accuracy of *ADSS* and *GCSM* method are *17%* and *33%* respectively . The results show that the performance of this article method is better than *GCSM* and *ADSS* because of considering the characteristics of instances multiple inheritance in the case of instance multiple inheritance relationship is complexer than the property relationship .

10. Conclusions

Semantic similarity calculation is a key technology based on the semantic web information retrieval . This paper analyzes instances multiple inheritance effects on instances similarity. For the original lack of instance semantic similarity calculation method , a comprehensive similarity calculation method of the hierarchy factors between instance property and prpperty values in multiple inheritance and ontology knowledge base is proposed. Experiments show that the proposed method in this paper can improve the accuracy of similarity in the case of the composition under a variety of ontology knowledge base.

Acknowledgments

This work was supported by the Social Science Foundation of the Colleges and Universities of Jiangsu Province (*No.2012SJD870001)* and the Science and Technology Research Project of Huai'an City (*No.HAG2012055*). The authors would like to express our sincere thanks to the editors and the reviewers for giving very insightful and encouraging comments.

References

[1]Y.X.Yu, "Semantic Information Retrieval Study Based on Knowledge Reasoning", Journal of Information (in Chinese) , Vol.27, No.11, 2008, pp.78-80.

[2]J.Jiang, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy", InProc,Research on Computational Linguistics, 1997, Vol.1, pp.19-33.

[3]P.Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy",InProc 14th Int'l Joint Conf, Artificial Intelligence, 1995, Vol.1, pp.48-53.

[4]B.Franz, C.Diego, M.G.Deborah, et al, The description logic handbook,India: Cambridge University Press, 2003, pp.189-212.

[5]Y.X.Yu, "Information query model based on OWL-S matching", Computers and Applied Chemistry (in Chinese), Vol.24, No.9, 2007, pp.1277-1280.

[6]P.G.Lin, H.Liu and X.Z.Fan, et al, "New method for query answering in semantic web", Journal of Southeast University (English Edition), Vol.22, No.4, 2006, pp.319-323.

[7]Y.X.Yu. and Y.Y.Yan, "Information Retrieval Method based on Page Segmentation", Library and Information Service, Vol.53, No.3, 2009 , pp.108-110,114.
[8]L.J.Zhu, L.Tao and H.Liu, "Caculation of the concept similarity on domain ontology", Journal of South China University of Technology (Natural Science Edition), Vol.32, No.1, 2004, pp. 147-159.
[9]Y.J.Liu and Y.Xu, "Automatic question answering system based on weighted semantic similarity model",Journal of Southeast University (Natural Science Edition), Vol.34, No.2, 2006, pp. 609-612.
[10]M.Rodriguez and M.Egenhofer,"Determining semantic similarity among entity classes from different ontologies", IEEE Transactions on Knowledge and Data Engineering, Vol.15, No.1, 2003, pp. 442-456.
[11]U.Straccia, "Reasoning within Fuzzy Description Logics",Journal of Artificial Intelligence Research,Vol.14, No.1, 2001, pp.323-328.
[12]Brian McBride.Jena, "A Semantic Web Toolkit", IEEE Internet Computing, 2002,Vol.6, pp. 55-59.
[13]Aleman-Meza B, SWETO, "Large-scale Semantic Web Test bed ", Proceedings of the 16th international Conference on Software Eng &Knowledge Eng (SEKE2004), Workshop on Ontology in Action, Banff, Knowledge Systems Inst, 2004, Vol.1, pp.490-493.
[14]Kerschberg.Larry, Kim.Wooju and Scime.Anthony, "A personalizable agent for semantic taxonomy-based web search", Lecture Notes in Artificial Intelligence, 2003, Vol.1,pp.13-31.
[15]Ganesan P, "Exploiting Hierarchical domain structure to compute similarity", ACM Transactions on Information System, Vol.21, No.3, 2003, pp.64-93.
[16]L.X.Han and L.P.Sun, "An approach to determining semantic similarity". Advances in Engineering Software, Vol.27, No.1, 2006, pp.129-132.

Yangxin Yu was born in Taizhou City of Jiangsu Province, China in 1970 and respectively received both his B.Sc degree(1995) and the M.Sc degree(2004) from College of Computer Science and Technology,Wuhan University of Technology and School of Computer Science and Technology, Suzhou University.

He is currently an associate professor in the Faculty of Computer Engineering, Huaiyin Institute of Technology,Information society member of Jiangsu Province and anonymous reviewers of Library and Information Service. His research covers a broad range of topics within AI including information management and information systems, information retrieval,intelligent information processing and personalization techniques.

Prof.Yu has authored or co-authored about 40 journal and conference papers, a few of which have got the honor of Huaian City Natural Science outstanding papers and has won the third prize of Science and Technology Progress Award of Huai'an City, edited three textbooks in chief.

Yizhou ZHang was born in Huaiyin of Jiangsu province, China in 1981 and respectively received both his B.Sc degree (2003) and the M.Sc degree(2010) from College of Computer Science and Technology, University of Science and Technology of China and

School of Computer Science and Technology,Nanjing University of Science and Technology.

He is currently a lecturer at Party School of Chinese Communist Party Huai'an City Committee, China. His research interests include multimedia systems, user modeling, information management and information systems,intelligent information processing and personalization techniques.

Mr.ZHang has authored or co-authored about 10 journal and conference papers, edited one textbook in chief.