Searching Concepts and Association Relationships Based on Domain Ontology

Kunmei Wen, Ruixuan Li
School of Computer Science and Technology
Huazhong University of Science & Technology
Wuhan, China
{kmwen, rxli}@hust.edu.cn

Bing Li
State Key Laboratory of Software Engineering
WuhanUniversity
Wuhan, China
bingli@whu.edu.cn

***Abstract*—This paper presents an ontology-based semantic search system called Smartch. This system allows users to search concepts and association relationships based on domain ontology. Compared to current search methods, Smartch provides a kind of user-defined graphical queries and a function of searching the association relationships that exist between two concepts or instances. It makes use of a novel ranking method that implements searching for concepts and association relationships. In this paper, we introduce the methods of ranking designed for Smartch. A case study in the academic domain is given in this paper to illustrate the functionalities of Smartch. This system has been developed and tested on the academic websites using an ontology that models the academic domain. The system can be extended to other domains. All of these capabilities allow the system to provide more intelligent functions than traditional search engines. The performance enables the use of the system in real environment.**

***Keywords*—semantic search, ranking, association relationship**

I. INTRODUCTION

As the use of the Internet has accelerated, search engines have more and more become a part of people's lives. Search engines can find useful and valuable information for users. However, traditional search engines still have some limitations. Most search engines are based on keyword queries and too many results are returned. Users have higher and higher demands for accuracy and intelligence in the work of search engines. Most search engines focus on finding related web pages based on keywords. Sometimes users want to know which instances belong to certain concepts and what kinds of relationships exist between two concepts or instances. Up to now some search engines have made the effort to search for concepts based on Ontology and Semantic Web [1] technology. One of the difficulties is how to provide a friendly interface for users. Users need a good way to express their query requirements.

Despite many applications using domain ontology in information retrieval, relatively few of them [2] are concerned with association relationship search. There is still little work dealing with research on the association relationships between two instances or concepts. Most research emphasizes concept search. Ranking the search results is the key technology of semantic search. Since it is expected that the number of relationships between entities in a Knowledge Base (KB) will be much larger than the number of concepts and instances themselves, the likelihood that association relationship search would result in an overwhelming number of results for users is increased, thereby elevating the need for appropriate ranking schemes.

In this paper, we design and implement a semantic search engine Smartch, which has different kinds of intelligent functions. It can find the concepts and association relationships that exist between two concepts or instances. The rest of the paper is organized as follows. Section 2 reviews related work on semantic search. The core of our paper, Section 3, proposes the approach that is used in Smartch. Section 4 shows the implementation of Smartch. Finally, in Section 5, we draw some conclusions and express our thanks.

II. RELATED WORK

Semantic search [3] integrates the technologies of Semantic Web and search engines to improve the search results gained by current search engines. It finds the semantic information by means of inferring the internal knowledge in KB. According to the role of ontology, Semantic search can be sorted into three types: enhanced semantic search based on the traditional search, semantic search based on ontology knowledge, and other forms of semantic search.

The first type of semantic search, enhanced semantic search based on the traditional search, makes use of semantic technology to improve traditional search results. Its core is still the traditional search engine, such as Tap [4].

The second type of semantic search, semantic search based on ontology knowledge, by reasoning provides the internal knowledge to users. According to the search object, we can divide this kind of semantic search into a concept search and an association relationship search. Concept search includes Swoogle [5].Association relationship search refers to a search for the complicated association relationships between two resources and then ranks them. The main problem with an association relationship search is how to measure the user's interest in the link path. Anyanwu and Sheth [6] present a simple formal and popular method to find the valuable association relationships that exist between two resources. Ranking the search results is the key technology of a semantic search. The ranking method [7] focuses on the semantic metadata to find the complicated association relationships and

978-0-7695-4313-0/10 $26.00 © 2010 IEEE
DOI 10.1109/GCC.2010.90

predict the user's need to distinguish different association relationships.

The last type of semantic search is the other form of semantic search. The KnowItAll system [8] developed by Turing Center is used to extract web information, with the purpose of building an artificial intelligence system. Semantic search makes the effort to derivate implicit knowledge. The literature [9] proposes a method of using spread activation technology to find related concepts in a given ontology. This method describes an algorithm to find relevant concepts, such as the author of a document. Text annotations are formed on an RDF graph. Hsieh et al [10] propose a query-based ontology knowledge acquisition system that dynamically constructs query-based partial ontology to provide proficient answers for users' queries. It focuses on building knowledge, not searching and ranking the concept and relationships. Meo et al [11] present a new approach that supports users in annotating and browsing resources referred by a folksonomy. It does not use the ontology-based method. In [12] user ontology is built to capture the users' interests in order to improve personalized Semantic Web searches.

The present work of semantic searching focuses on concept search. One of the difficulties is how to provide a friendly interface for users. Users need a good way to express their query requirements. At the same time, there is less work on how to express a user's query especially through graphic-defined queries. In this paper we have explored two points. The first deals with a kind of user-defined graphical query. The second deals with association relationship search and rankings. We provide a method for ranking the association relationships comprehensively by evaluating the weight of the resource.

III. ARCHITECTURE AND METHODS

A. *Architecture of semantic search engine Smartch*

Here we present the architecture of the semantic search engine Smartch. It is shown in Figure 1. The components and the relationships between them are described as follows:

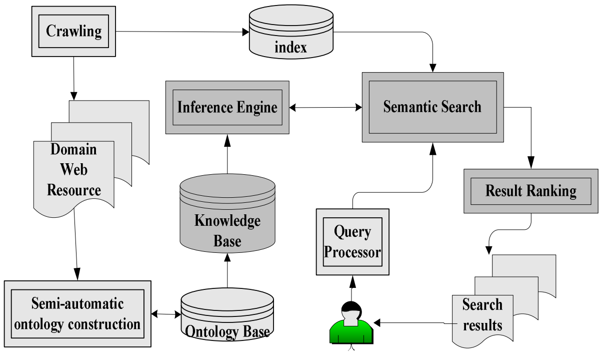


Figure 1. Architecture of the semantic search engine Smartch

The crawling component collects the related web pages from the domain web resource. And then the crawled web pages are indexed. The Query Processor receives queries from users. The query is defined as keywords or formal queries or user-defined graphic queries. The Query Processor converts the user's queries into a uniform format, which is defined by the semantic search engine Smartch. Then these queries will be distributed in two ways. One is forwarded to a traditional search engine and obtains information from the index. The other is forwarded to an inference engine. By means of the operation of a traditional search engine, we will get the initial results using text IR technology. The initial search results are also transformed to the inference engine. If the user submits a formal query, then the query will push on directly to the inference engine. The ontology is built in a semi-automatic manner. And the ontology is transformed into KB. KB restores the domain ontology and reasoning rules or knowledge and is the base for reasoning. The inference engine performs the reasoning operations to get the semantic information and obtains all the search results. The Result Ranking Engine ranks all the results returned by the inference engine. Finally the user gets the final results through result ranking.

B. *Query definition*

Four types of search are provided by Smartch, namely basic search, concept search, user-defined concept search and association relationship search.

Q_{i1} is a basic search, formed as $Q_{i1} = "A"$ where A means a keyword that appears in the web pages, and the returned results are the pages containing the keyword and its synonyms. In fact, Q_{i1} is similar to traditional query.

Q_{i2} is a concept search, formed as $Q_{i2} = "A"$ where A means a concept. Returned results are all the instances that belong to the concept A .

Q_{i3} is a user-defined concept search where the user can define the query in a graphic manner. First, one ontology concept is selected. Then all the properties of the concept are shown to be extended. The user clicks the selected property to expand the query graph and to restrict it. Again the process can be circular until the user-defined process is over. Finally, the user needs to set the query concept.

Q_{i4} is an association relationship search, formed as $Q_{i4} = (O_1, O_n)$ where both O_1 and O_n are entities including concepts and instances. The association relationships between O_1 and O_n are returned.

C. *Ranking methods*

Ranking search results is very important for the realization of a semantic search. Smartch provides three different ranking methods for different queries. These methods are the following: ranking web pages, ranking concepts, and ranking association relationships. The key problem is how to rank the relationships between concepts. We provide a ranking scheme based on the ranking weight.

1) *Ranking web pages*

For the basic search, given the query Q_{i1}, the results are web pages $W_1, W_2, ..., W_{i,} ......, W_n$.

The ranking weight of the web page W_i is defined as R_i . Here the tf/idf [13] method is used to calculate the ranking weight R_i for the resultant web page W_i .

2) Ranking concepts

For the concept search and the user-defined concept search, given the query Q_{i2} and Q_{i3}, the results are instances $I_1, I_2, ..., I_{i,} ......, I_n$, which belong to the query concept. The instance I_i is treated as a keyword in performing the basic search. For I_i the number of I_i's related web pages is n_i. n_i is used as the ranking weight. It means that the more related pages, the more important the instance.

3) Ranking association relationships

For the association relationship search, given the query Q_{i4}, the results are relationships as follows:

$$R_i = \{O_1, P_{1i}, O_{2i}, P_{2i}, O_{3i,} ......, O_{(n-1)i}, P_{(n-1)i}, O_n\}, i = 1, 2, ..., m$$

The ranking weight R_i is determined by the importance weight of the relationship. We propose a method to calculate the ranking weight of the relationships.

First, we need to define different relationship impact factors. Three different impact factors are defined, namely the domain relevance, the length relevance, and the frequency relevance.

Definition 1. Domain Relevance, expressed as D_R, is defined as the relevance between the entities or attributes in the relationship R and the domain in which the user is interested.

Users may be more interested in the relationships in a given special domain. Different users have different interest domains. Here we assume that the domain in which the user is interested is D. So the entity or attribute set of the relationship R that belongs to domain D is the following:

$$Y_i = \{O_i \ or \ P_i \mid O_i \in R \cap P_i \in R \cap O_i \in D \cap P_i \in D\}$$

The entity or attribute set of the relationship R that does not belong to domain D is the following:

$$N_i = \{O_i \ or \ P_i \mid O_i \in R \cap P_i \in R \cap O_i \notin D \cap P_i \notin D\}$$

For example, the academic domain contains the academic-related concepts and attributes, generally including entities such as "teacher", "paper", "course", and attributes such as "publish", "teach". The domain relevance of a relationship is calculated by the formula (1):

$$D_R = d + (1-d) \times \frac{|Y_i|}{length(R)} \times (1 - \frac{|N_i|}{length(R)}) \quad (1)$$

Here, $length(R)$ denotes the path length of the relationship R, where d is an adjustment factor that is set in order to avoid $D_R = 0$. The size of d is generally between 0 and 1. The formula (1) shows that the domain relevance is directly proportional to the number of entities and attributes of the relationship R , which belongs to the user's interested domain D.

Definition 2. Length Relevance, expressed as L_R, is the impact factor, which measures how the path length of the relationship impacts the results ranking.

The query result is $R = \{O_1, P_1, O_2, P_2, O_{3,} ......, O_{n-1}, P_{n-1}, O_n\}$. n is the path length. Under normal circumstances, a shorter path length indicates a higher relevance; in some cases, however, the opposite is true.

$$L_R = \frac{1}{length(R)} \ or \ L_R = 1 - \frac{1}{length(R)} \quad (2)$$

Formula (2) gives two methods for calculating the length relevance. The former shows that the shorter path length, the greater the length relevance. The second method of calculation is exactly the opposite. Users can choose a suitable formula to calculate the length relevance combined with the actual requirements.

Definition 3. Frequency Relevance, expressed as F_R, is the impact factor that measures how the degrees of the entities of a relationship impact the results ranking. In this case, the entity could be a concept or instance.

Similar to PageRank [14] technology, an entity with a greater degree has a higher relevance. The frequency relevance is determined by the in-degree and out-degree of the entities of a relationship. Formula (3) gives an approach for calculating the frequency relevance.

$$F_R = \frac{I_R + C_R}{2} \quad (3)$$

Here, I_R is the in-degree of the relationship R, C_R is the out-degree of the relationship R. If the number of entities in the relationship R is n, the method of calculating I_R and C_R is $I_R = \frac{1}{length(R)} \sum_{i=1}^{n} \frac{I_i}{Max(I)}$, where I_i is the in-degree of the entity O_i, $Max(I)$ is the biggest in-degree of n entities. $C_R = \frac{1}{length(R)} \sum_{i=1}^{n} \frac{C_i}{Max(C)}$, where C_i is the out-degree of the entity O_i, $Max(C)$ is the biggest out-degree of n entities. So the formula (4) is an equation as:

$$F_R = \frac{1}{2length(R)} \sum_{i=1}^{n} (\frac{I_i}{Max(I)} + \frac{C_i}{Max(C)}) \quad (4)$$

The final ranking results of relationships need to be considered for various factors. Based on the three key impact factors calculated above, a ranking method is proposed as follows. For the query $Q = (O_1, O_n)$, the search results are relationships:

$R_i = \{O_1, P_{1i}, O_{2i}, P_{2i}, O_{3i,} ......, O_{(n-1)i}, P_{(n-1)i}, O_n\}, i = 1, 2, ..., m$. In this method, the importance weight of the relationship R is calculated by the formula (5):

$$V_R = k_1 \times D_R + k_2 \times L_R + k_3 \times F_R \quad (5)$$

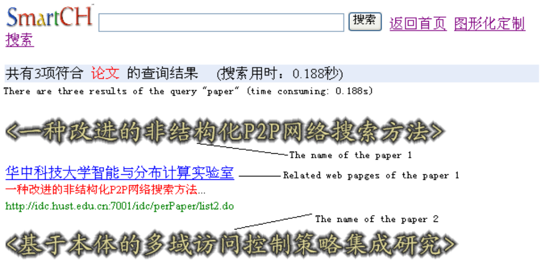
Where $k_1 + k_2 + k_3 = 1$, different users have different requirements for ranking. Users can assign k_1, k_2, k_3 it according to their actual requirements. The value D_R denotes domain relevance. The value L_R denotes length relevance. The value F_R denotes frequency relevance.

Finally, the relationship results of the query will be ranked according to the value of the importance weight V_R . The greater V_R indicates that the relationship is more important and will be a priority to the user.

IV. SYSTEM IMPLEMENTATION

The implementation of the semantic search Smartch is based on the integration of search and inference. First, we construct the domain ontology. Then we use Lucence as an Intranet search tool and Jena as an ontology parsing tool. To improve the efficiency for the operation of uploading OWL files, SQLServer is used as the tool for regularly storing the ontology data. The ontology model can be read directly from the database. We use Pellet, which is an open-resource reasoning tool, as the reasoning engine.

Basic search and Concept search are realized in semantic search Smartch. If the user chooses to run a concept search, then the instances of the concept are returned. For example, when "paper" is submitted as the query of concept search, the result is shown in Figure 2. All the instances that belong to the concept "paper" are returned.


SmartCH 搜索 返回首页 图形化定制搜索

共有3项符合 论文 的查询结果 (搜索用时：0.188秒)

There are three results of the query "paper" (time consuming: 0.188s)

<一种改进的非结构化P2P网络搜索方法>

The name of the paper 1

华中科技大学智能与分布计算实验室

Related web papges of the paper 1

一种改进的非结构化P2P网络搜索方法...

http://idc.hust.edu.cn:7001/idc/perPaper/list2.do

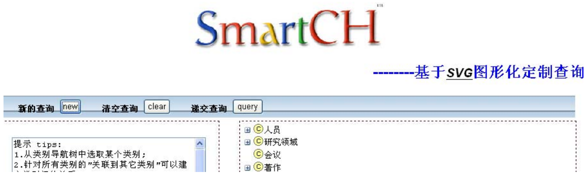
The name of the paper 2

<基于本体的多域访问控制策略集成研究>

Figure 2. The result of concept search

A. *User-defined graphical concept query*

For the research in the present paper, a user-defined process of graphic query was implemented based on SVG (Scalable vector graphics). SVG graphic technology and Ajax are adopted on the client. Grahpviz is the tool used to generate a SVG format string flow. The popular concepts are shown on the right side of Figure 3.


SmartCH

--------基于SVG图形化定制查询

新的查询 new 清空查询 clear 递交查询 query

提示 tips:
1.从类别导航树中选取某个类别；
2.针对所有类别的"关联到其它类别"可以建

⊞ ©人员
⊞ ©研究领域
©会议
⊞ ©著作

Figure 3. Concept selecting of user-defined query

Then the user chooses the query variable. Some constraints are added on the query variable shown in Figure 4. In this example, the concept "associate profess" is defined as a query variable. Then some constraints are added to it, including teaching the course "Database", guiding the student "Ji Yong", and born in "Province Hubei" .

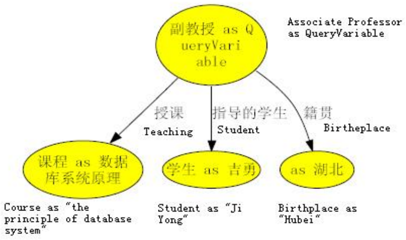
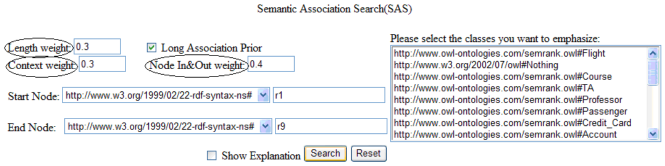


Figure 4. The completed user-defined query

B. *Association relationship ranking*

If the user inputs two entities, the implication is that the user wants to find the relationship between them. In Smartch, we input "r1" and "r2". The system returns the relationship that exists between "r1" and "r2". The interface is shown in Figure 5.


Semantic Association Search(SAS)

Length weight: 0.3 ☑ Long Association Prior

Context weight: 0.3 Node In&Out weight: 0.4

Start Node: http://www.w3.org/1999/02/22-rdf-syntax-ns# r1

End Node: http://www.w3.org/1999/02/22-rdf-syntax-ns# r9

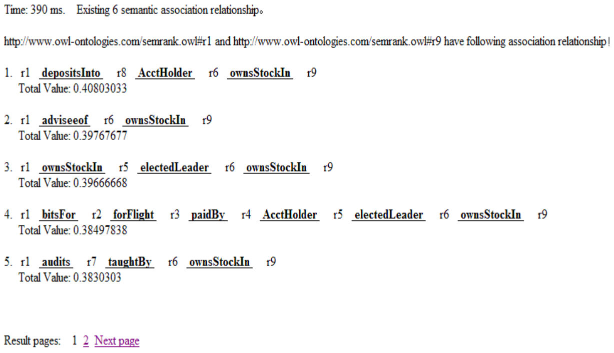
Please select the classes you want to emphasize:
http://www.owl-ontologies.com/semrank.owl#Flight
http://www.w3.org/2002/07/owl#Nothing
http://www.owl-ontologies.com/semrank.owl#Course
http://www.owl-ontologies.com/semrank.owl#TA
http://www.owl-ontologies.com/semrank.owl#Professor
http://www.owl-ontologies.com/semrank.owl#Passenger
http://www.owl-ontologies.com/semrank.owl#Credit_Card
http://www.owl-ontologies.com/semrank.owl#Account

□ Show Explanation Search Reset

Figure 5. The input window of the association relationship search

Three phrases are highlighted with ovals on the input window. They are Length weight, Context weight, and Node In&Out weight. These three weights respectively indicate the length relevance, the domain relevance, and the frequency relevance defined in section 3. Their range is between 0-1, and their sum is equal to 1. The other phrase is Long Association. If Long Association is selected, the association relationships with longer paths are ranked first, or the shorter ones are ranked first.

The search results are shown in Figure 6. Examining the results, we find that there are five relationships that exist between "r1" and "r2". For each result, the ranking score is calculated by using the method given above.


Time: 390 ms. Existing 6 semantic association relationship。

http://www.owl-ontologies.com/semrank.owl#r1 and http://www.owl-ontologies.com/semrank.owl#r9 have following association relationship!

1. r1 depositsInto r8 AcctHolder r6 ownsStockIn r9
   Total Value: 0.40803033

2. r1 adviseeof r6 ownsStockIn r9
   Total Value: 0.39767677

3. r1 ownsStockIn r5 electedLeader r6 ownsStockIn r9
   Total Value: 0.39666668

4. r1 bitsFor r2 forFlight r3 paidBy r4 AcctHolder r5 electedLeader r6 ownsStockIn r9
   Total Value: 0.38497838

5. r1 audits r7 taughtBy r6 ownsStockIn r9
   Total Value: 0.3830303

Result pages: 1 2 Next page

Figure 6. The result of the relationship between "r1" and "r2"

Currently, there is no commonly agreed upon evaluation methodology and benchmark for semantic search. We ourselves constituted our own research group's evaluation dataset. The results have been analyzed positively. The dataset is made up of the academic ontology and the set of campus web pages (more than 200MB). We can use the traditional test method of search engines. The proposed ranking method can be evaluated by comparing the real values with the expected values.

1) Testing using single ontology

The ontology *semrank.owl* is used as the test case. The ontology is involved in several domains with some complicated relationships. Five typical combination queries are defined. Every query is designed to test two ranking impact factors. That means the two factors are given higher weights. Five participants carry out the test. Taking into account the users' subjectivity, the average ranking results are defined as the ideal ranking values. The intersection results of system ranking and ideal ranking are shown in figure 7. The results imply the consistency of system ranking and ideal situation. The results indicate that the results of system ranking are close to ideal ranking, some sort results directly match.

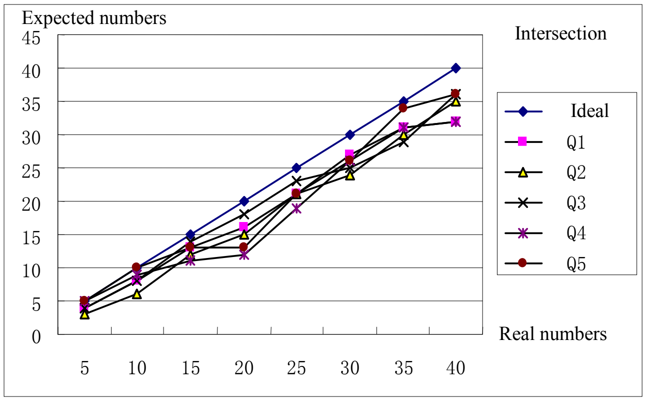


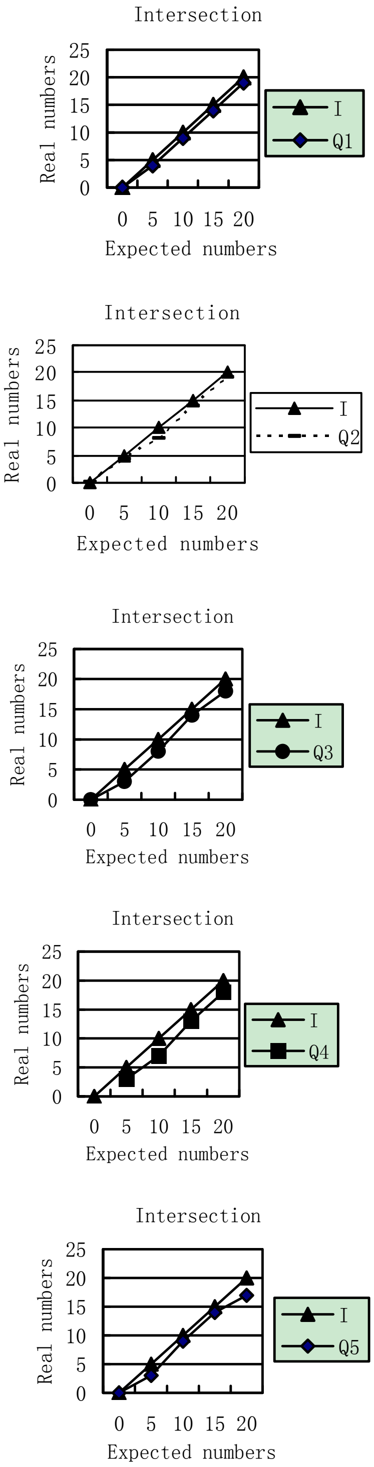
Figure 7. Test results of RAR

According to the precision formula: $P = \frac{|R_a|}{|R|}$, here $|R_a|$ is the number of retrieved documents which are related to the query, $|R|$ is the total number of related documents. Therefore the average precisions of Q1, Q2, Q3, Q4, Q5 are 83.6%, 76.5%, 86.4%, 81.8%, 88.7% respectively. The total average precision is 83.4%. The results indicate the average precision is higher than 80%. The bias of user query results and expectations is within an acceptable range.

2) Testing using multi ontologies

The ontologies *Semrank.owl, Animal.owl, Idc_onto.owl , Apex_Portal_0.99.owl* are used as test cases. Users randomly select queries. The average results are evaluated. For each association query the top 20 results are selected. Eight combinations of impact Factors for ranking are given. For each possible query, before the test we give the ideal association relationship ranking results manually. The designers of the ideal results and the experimental test are not the same person. They are both the students of computer science who are familiar to relationship search.

Q1 and Q4 are single-factor tests. Q5 and Q8 are multi-factor tests. Figure 8 indicates the test results of Q1-Q8. I represents the ideal value. To test the effectiveness of the ranking method, expectation and the actual results are given in the figure 8. The comparison figure 8 shows the intersection number of user expectations with the actual results of the relationship queries. Ideal query is a kind of ideal situation which indicates that the expected values are totally consistent with the real values.



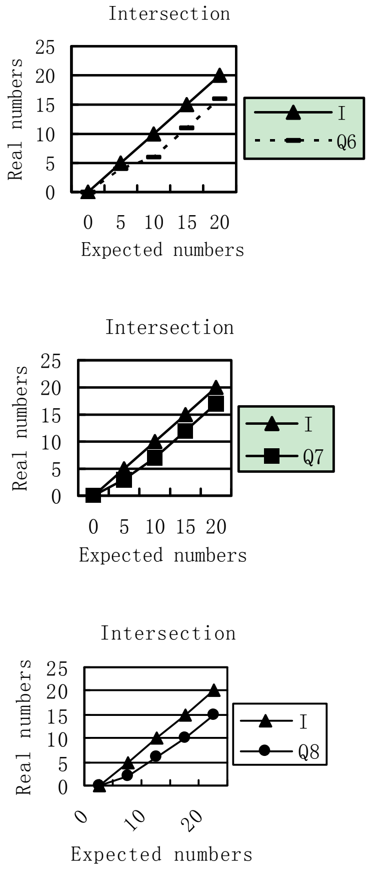


Figure 8. Testing results of Q1- Q8

The results show that the single factor test has the better performance than the multi-factor test. For the multi-factor test different users have different criteria, so there are some deviations. Based on average precision formula $\overline{P} = \frac{1}{8}\sum_{i=1}^{8} P_i$, where P_i the corresponding precision of query Qi, the average ranking precision is higher than 80%. There is an acceptable deviation between the user query results and the ideal conditions. The ranking method can first return the expected relationship results to users. Although the differences of the users' ranking standards, the test results shows the feasibility of the ranking method, which can meet the various preferences of different users and allows users to obtain satisfactory results.

V. CONCLUSIONS

Semantic search is different from traditional search. They use semantic search technology to improve the search results. We have developed a semantic search engine named Smartch. The experiment shows that Smartch can improve recall, through keyword parsing based on ontology. It can also extend a keyword to its equivalent concept and sub-concept. A concept query searches all the instances of the concept through inference. The user-defined graphical method can inerrably return the semantic information hidden in a user's query and can improve precision. Smartch can also find the association relationship between two entities. It can implement some intelligent functions compared with traditional search engines. We have combined text IR with semantic references in Smartch. The system can be extended to other application domains. It can also answer complicated queries such as the relationships between two concepts or instances.

ACKNOWLEDGMENT

This paper is supported by National Natural Science Foundation of China under Grant No. 60873225, 60773191, 70771043, 60873083, Natural Science Foundation of Hubei Province (NSF-HB) under grant No. 2009CDB298, for Distinguished Young Scholars under grant No. 2008CDB351, Open Foundation of State Key Laboratory of Software Engineering under Grant No. SKLSE20080718 and Innovation Fund of Huazhong University of Science and Technology under Grant No. Q2009021.

REFERENCES

[1] T.Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, May 2001

[2] Anyanwu, K., Maduko, A., and Sheth, A.P.: SemRank: Ranking Complex Relationship Search Results on the Semantic Web, Proceedings of the 14th International World Wide Web Conference, ACM Press, 2005

[3] Guha R, McCool R, Miller E. Semantic search. Proceeding of the 12th International World Wide Web Conference (WWW 2003). Budapest, Hungary, May 2003: 700-709

[4] Guha, R., McCool, R.: TAP: A Semantic Web Test-bed. Journal of Web Semantics, 2003, 1(1)

[5] Ding L , Finin T, Joshi A, et al. Swoogle: A search and metadata engine for the semantic web. In CIKM'04. Washington DC, USA, November 2004

[6] Anyanwu, K., Sheth, A.P.: □-queries: enabling querying for semantic associations on the semantic web. In: Proceedings of the 12th international conference on World Wide Web, 2003: 690–699

[7] Boanerges Aleman-Meza, Christian Halaschek-Wiener, I. Budak Arpinar, Cartic Ramakrishnan, Amit P. Sheth, Ranking Complex Relationships on the Semantic Web, IEEE Internet Computing, 2005, 9(3): 37-44

[8] D. Downey, O. Etzioni, and S. Soderland. A Probabilistic Model of Redundancy in Information Extraction. Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI) 2005

[9] Cristiano Rocha, Daniel Schwabe, Marcus Poggi de Aragão: A hybrid approach for searching in the semantic web. WWW 2004: 374-383

[10] Tung Cheng Hsieh, Kun Hua Tsai, Ti Kai Chiu, Tzone I Wang, Ming Che Lee. Partially constructed knowledge for semantic query. Expert Systems With Applications, Volume 36, Issue 6, Pages 10168-10179, 2009

[11] Pasquale De Meo, Giovanni Quattrone, Domenico Ursino. Exploitation of semantic relationships and hierarchical data structures to support a user in his annotation and browsing activities in folksonomies. Information Systems. Volume 34 , Issue 6, Pages 511-535, 2009

[12] Xing Jiang, Ah-Hwee Tan. Learning and inferencing in user ontology for personalized Semantic Web search. Information Sciences, Volume 179, Issue 16, Pages 2794-2808, 2009

[13] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. Introduction to information retrieval. Cambridge University Press, 2008

[14] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, Sriram Raghavan. Searching the web. Volume 1, Issue 1, Pages 2 – 43, 2001