Journal of Southeast University (English Edition) Vol. 23, No. 3, pp. 357 – 360 Sept. 2007 ISSN 1003—7985

Domain semantic mapping of database metasearch engine

Miao Guangxiang[1] Chen Xiangyang[2]

([1] Baoding Technical College of Electric Power, Baoding 071051, China)

([2] College of Mathematics and Computer, Hebei University, Baoding 071002, China)

Abstract: In order to implement semantic mapping of database metasearch engines, a system is proposed, which uses ontology as the organization form of information and records the new words not appearing in the ontology. When the new word's frequency of use exceeds the threshold, it is added into the ontology. Ontology expansion is implemented in this way. The search process supports "and" and "or" Boolean operations accordingly. In order to improve the mapping speed of the system, a memory module is added which can memorize the recent query information of users and automatically learn the user's query interest during the mapping which can dynamically decide the search order of instances tables. Experiments prove that these measures can obviously reduce the average mapping time.

Key words: ontology; domain mapping; database metasearch engine; memory module

With the increase in web resources, various search engines have appeared in order to improve the efficiency of searching. A recent survey indicates that there are about 350 000 web databases (WDB, i. e. database search engine)[1]. The WDB has a back database which stores structured-data and a complex search interface (such as Amazon, Expedia and Realestate. com). With the rapid development of E-commerce, the study of WDBs becomes increasingly important, because most E-commerce search engines are WDBs.

Recent research efforts on the WDB have been mainly directed to constructing the DMSE system for WDBs of the same domain. They include how to cluster the WDBs related to the same domain[2], and to integrate the search interfaces of all the WDBs in the same domain into an integrated metasearch interface[3–7], by which we can compare the products from multi-WDB sites by one query, support distributed top-n query processing[8] and extract correct search result records from result pages returned by WDBs[9]. These have been tested by experiments proved to be effective. However, there are still many problems in the DMSE system.

Domain mapping is one of them. When the number of metasearch engines of different domains reaches hundreds of thousands, it is not feasible for users to select the needed metasearch engine by themselves. Therefore, it is necessary to add a mapping module which can map the user's query into the required metasearch engine and then carry out the detailed query. However, since users do not know to which domain the query product belongs and that even for the same domain query inputs also vary, it becomes a difficult problem as how to implement domain mapping. The domain mapping module should be able to map the different query inputs related to the same domain into one domain by semantic information. References to the study of this problem have not been seen so far.

In this paper we study domain semantic mapping, the main aspects involved in mapping information ontology organizing, the choosing of the ontology database during queries, the amplification of the ontology and supporting the query Boolean operation.

1 Ontology-Based Domain Mapping Study

1.1 Construction of ontology

1.1.1 Problem definition

According to the ontology definition of Gruber, ontology is an explicit specification of a conceptualization[10]. The aim of ontology is to define the relationships between terms within a certain domain and describe and represent the domain knowledge in order to facilitate the data's automatic processing.

Due to the following need for the construction of the ontology, we provide the following components of the ontology: ① Concepts: which can be anything, e. g. automobile and brand. ② Attributes: the description of one aspect of concepts, e. g. the concept "price" is an attribute of the concept "automobile". ③ Concept relationships: the interaction of concepts in one domain, e. g. the relationship of the concept saloon car and automobile is kind-of. ④ Instances of concept: i. e. concept entities, e. g. "Santana" is the instance of the con-

Received 2007-05-18.

Biography: Miao Guangxiang (1974—), male, graduate, miaoguangxiang@ sina. com.

cept "brand". ⑤ Synonyms: the meanings of concepts are identical, e. g. the concept "make" may also be denoted by "brand", "car manufacturer", etc. ⑥ Broader term: term A is a broader term of term B if the connotation of term A includes term B, e. g. "fruit" is a broader term of "apple".

1.1.2 Construction and storage of ontology

Presently, the three ways of constructing the ontology are manual, semi-automatic and automatic. Since the semi-automatic and automatic ways are based on the analysis and statistics of training documents, the accuracy of the ontology is not high and manual modifications are often needed. Moreover, there are many technical difficulties[11]. Therefore, in this paper the manual way is adopted to construct the ontology. Though expert participation is often needed in the construction of the ontology, this paper only involves five domains, which are automobile, household appliance, book, shoes and cosmetic, and is only concerned with the kind-of, instance-of, attribute-of, broader-of and synonym relationships between concepts, not the knowledge of the field as a whole. So we just refer to the existing domain ontology and dictionaries to construct the needed ontology.

Before studying the way of the storage of the ontology, we will first analyze the specific ontology. Fig. 1 is an instance of an ontology. As can be seen, the conceptual relationships in the ontology are simple and can be expressed as {concept, attribute and instance}. Since there is some relationship between domain concepts (e. g. the relationship of the concept saloon car and automobile is kind-of), and the relationship of the attribute of concept and its instances is $1:n$, so it is necessary to respectively store domain concepts and instances in different tables of a relational database. Different domains may be related to different attributes, therefore, when the attributes of one domain are different from one another, their instances should be stored respectively. Thus we need a table to store domain concepts and several tables to store instances.

1.2 Memory module

If every query is matched in all the ontology databases during the query mapping, the efficiency of the system will be reduced. For more frequent queries, the mapping speed should be quicker. Usually, the human brain extracts the frequently used-information from memory more quickly than the seldom used-information. Psychological studies on the extraction of long time memory show that the more frequently information is used, the less its threshold of activation is. Based on the studies, we have designed a memory module to simulate the extraction of memory of human brain.

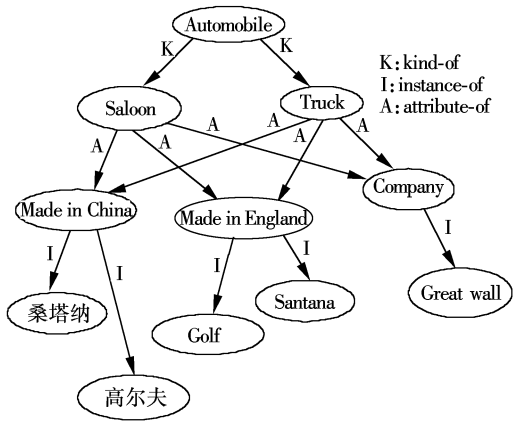

Fig. 1 The example of ontolgy

The manifestation of the memory module is a database table in this paper. In order to simulate the extraction of memory of human brain, the size of the table cannot be too large. Since the query time is explicit when the records are more than ten thousand, the memory table is not to exceed ten thousand records in this paper. The original records of the table can be obtained by learning. Every query is first searched in the memory table, and if the query record can be found in it, add one to its count field. Otherwise, the search should be in the ontology database, and if the query record is found, we append it to the memory table. All the record count fields automatically subtract one at a week's interval, which simulates memory fading of the human brain. When the table is full, the record whose count field value is minimal will be replaced. Because the size of the memory table is rather small and the records are all those that are recently used, the speed of searching frequently used-information is very fast. Thus, it not only improves the query efficiency of the system but makes the system intelligent.

1.3 Query selection of ontology database

When the user's query needs to be mapped into a specific domain, the matching first begins with the memory table, then with the domain table, and then with the instances tables. If the number of instances tables of domains reaches hundreds of thousands, it is worthwhile to consider the question of which order the search in the instances tables should be accorded.

Therefore, this paper adopts automatic implicit learning to learn the user's query interest. First, create an empty query interest table to store the domain name and interest degree. During the first query, randomly decide the search order of instances tables. If the needed domain is found, we record the information not only in the memory table but also in the query interest table. When the interest table is not empty, the search order of instances tables is decided by the interest degree of domains. The higher the interest degree is, the

earlier the search order of the domain's instances table is. For the domains which do not appear in the interest table, their instances tables are ordered after the tables that have been randomly ordered by interest degree. With the increase in query times, the information in the interest table becomes more abundant and more accurately represents the user's interest. When the user's interest changes as time goes by, the degree of user's interest should decrease to adapt to the change in the user's interest.

1.4 Ontology expansion

Because the establishment of the ontology is not complete, many new concepts need to be appended to it as the system is used. Therefore, we have created a table for new words. If the query word cannot be found in the ontology database, then append it to the new words table. The use frequency of every word in the table should be checked periodically, and when the frequency of a word is above a certain threshold, administrators or experts decide the domain to which the word belongs, and append it to the ontology database.

2 Boolean Operations Support of Query

In order to achieve accuracy, users often input several words at the same time. Therefore, segmentation of words is needed while query maps. The present system first uses the existing method of word segmentation to deal with user query inputs, then simultaneously maps the processed words, and supports "and" and "or" Boolean operations for these words.

When one of the query words is mapped to several domains, the system does the "and" operation and the result is returned to the user. For example, the input "great wall automobile" will become two words "great wall" and "automobile" after word segmentation. If "great wall" is mapped to "automobile" and "computer" domains, the mapping domains of "great wall" and "automobile" does the "and" operation and return the "automobile" domain to the user. When some of the query words cannot be found in the ontology database, the "or" operation is used between the mapping domains of the query words.

3 Experiments

In order to analyze the effect of constructing a user's query interest and memory module on the query speed during query selection from the ontology database, we designed two contrastable experiments in this paper. Because what experiments contrast is what effect different selection order of domain instances tables has on the query time. The words used in the query in the experiments are all the instances from various domains. Five domain instances tables are used in the experiments, and the records number of every instances table are all between 2 000 and 4 000. In Fig. 2, the stars indicate the search results of query according to the given order, while the circles in Fig. 2(a) indicate the search results of query according to the memory module and the carmine circles in Fig. 2(b) indicate the search results of query according to user interest. Every point in the figures shows the average search time of every five queries.

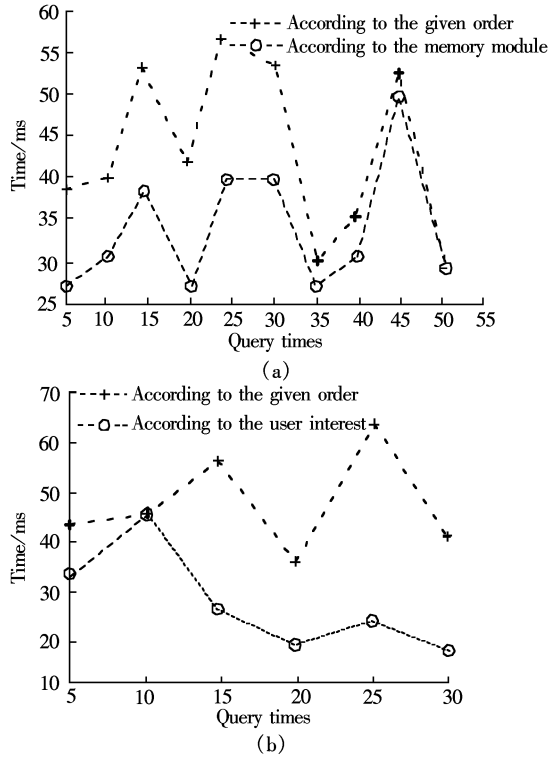

Fig. 2 Comparison curves of query time

In the contrastable experiments of different selection orders for domain instances tables, Fig. 2(a) shows the contrastable experimental results of introducing the memory module and according to the given order to select the domain instances tables. The number of records in the memory table is more than 2 000. Fifty queries have been performed in the experiment and 50 different query words are used, thirty of which appear in the memory table, accounting for 60%. The query order of the 50 words is decided randomly during the query. Then queries are done according to the same order under the memory module to compare results. It can be seen from Fig. 2(a) that the average query time has distinctly decreased after the introduction of the memory module. This is because the domains of the query words in the memory table can be directly determined so it does not need to search in the instances tables. Furthermore, when the number of records in the memory table increases, the probability of query words that

appear in the memory table also increases, thus the system average query time can decrease more.

Fig. 2(b) shows the experiment results of, respectively, query by the user's interest order and query by the given order. Altogether 30 queries are made in the experiment. It can be seen from Fig. 2(b) that the average time used by query according to the user's interest order is shorter than that used according to the given order. What's more, with the increase in queries, the average time greatly decreases. That is because with the increase in queries, user's query interest is also constructed. The user's interest decides the selection order of instances tables, which can better reflect the user's needs, and can find the needed query domain faster.

4 Conclusion and Future Study

This paper studies the domain mapping problem of the DMSE system. It implements semantic mapping by using ontology as the organization form of mapping information, and adds a memory module to learn the user's interest to improve the efficiency of the system. Experiments show that these measures reduce the average query time. Furthermore, ontology expansion and Boolean operations support are also dealt with.

The future studies can be on the query passing and result receiving to every WDB, extracting and merging results and so on.

References

[1] Chang K C-C, He B, Li C, et al. Structured databases on the web: observations and implications[J]. *SIGMOD Record*, 2004, **33**(3): 61 – 70.

[2] Peng Q, Meng W, He H, et al. WISE-cluster: clustering e-commerce search engines automatically[C]//*Proc of the 6th Annual ACM International Workshop on Web Information and Data Management*. Washington, DC: ACM Press, 2004: 104 – 111.

[3] He B, Chang K. Statistical schema matching across web query interfaces[C]//*Proc of the 22nd International Conference on Management of Data*. San Diego, California, USA, 2003: 217 – 228.

[4] He H, Meng W, Yu C, et al. Wise-integrator: an automatic integrator of web search interfaces for e-commerce[C]//*Proc of the 29th International Conference on Very Large Data Bases*. Berlin: Morgan Kaufmann, 2003: 357 – 368.

[5] Wu W, Yu C, Doan A, et al. An interactive clustering-based approach to integrating source query interfaces on the deep web[C]//*Proc of SIGMOD*' 2004. Paris, France, 2004: 95 – 106.

[6] He H, Meng W, Yu C, et al. WISE-Integrator: a system for extracting and integrating complex web search interfaces of the deep web[C]//*Proc of the 31st International Conference on Very Large Data Bases*. Trondheim, Norway, 2005: 1314 – 1317.

[7] Dragut E, Wu W, Sistla P, et al. Merging source query interfaces on web databases[C]//*Proc of ICDE*' 2006. Atlanta, GA, USA, 2006: 46.

[8] Yu C, Philip G, Meng W. Distributed top-*n* query processing with possibly uncooperative local systems[C]//*Proc of the 29th International Conference on Very Large Data Bases*. Berlin: Morgan Kaufmann, 2003: 117 – 128.

[9] Zhao H, Meng W, Wu Z, et al. Fully automatic wrapper generation for search engines[C]//*Proc of WWW*' 2005. Chiba, Japan, 2005: 66 – 75.

[10] Gruber T. A translation approach to portable ontology specifications[J]. *Knowledge Acquisition*, 1993, **5**(2): 199 – 203.

[11] Ding Shengchun, Cen Yonghua, Gu Defang. Research on semantic retrieval based on ontology [J]. *Journal of the China Society for Scientific and Technical Information*, 2005, **24**(6): 702 – 707. (in Chinese)

基于本体的数据库元搜索引擎域映射

苗广祥[1]　　陈向阳[2]

([1] 保定电力职业技术学院,保定 071051)

([2] 河北大学数学与计算机学院,保定 071002)

摘要:为了实现数据库元搜索引擎在语义层次上的映射,采用本体作为信息的组织形式,并记录本体中没有的新词,当使用新词的频率超过一定阈值时,加入到本体库中,对本体进行扩充. 查询过程支持"与"和"或"布尔运算. 为了提高系统的映射速度,特别添加了一个记忆模块,记录用户近期的查询行为. 并在映射过程中自动学习用户的查询兴趣,以此动态决定实例表的查询顺序. 实验证明这些方法可以显著降低系统平均映射时间.

关键词:本体;域映射;数据库元搜索引擎;记忆模块

中图分类号:TP18