Journal of Southeast University (English Edition) Vol. 21, No. 1, pp. 1 - 5 Mar. 2005 ISSN 1003—7985

Approximate querying between heterogeneous ontologies based on association matrix

Kang Dazhou[1] Xu Baowen[1,2] Lu Jianjiang[1,2,3] Wang Peng[1]

([1]Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)
([2]Jiangsu Institute of Software Quality, Nanjing 210096, China)
([3]Institute of Science, PLA University of Science and Technology, Nanjing 210007, China)

Abstract: An approximate approach of querying between heterogeneous ontology-based information systems based on an association matrix is proposed. First, the association matrix is defined to describe relations between concepts in two ontologies. Then, a method of rewriting queries based on the association matrix is presented to solve the ontology heterogeneity problem. It rewrites the queries in one ontology to approximate queries in another ontology based on the subsumption relations between concepts. The method also uses vectors to represent queries, and then computes the vectors with the association matrix; the disjoint relations between concepts can be considered by the results. It can get better approximations than the methods currently in use, which do not consider disjoint relations. The method can be processed by machines automatically. It is simple to implement and expected to run quite fast.

Key words: semantic web; information retrieval; ontology; query; association matrix

Information retrieval and filtering[1] is one of the most basic and important services on the web. Ontology-based information systems on the semantic web can retrieval pages that refer to precise concepts but not ambiguous keywords, and validate them using logical reasoning[2]. This can greatly increase the precision and recall of queries compared with current techniques. However, ontologies face heterogeneity problems[3]. Different systems may use different ontologies, and cannot access each other directly. The queries have to be rewritten to suit the specified system[4]. The rewriting process replaces concept names in the query by concept names in another ontology. Since often no perfectly corresponding ontology exists, it requires approximation mechanisms. One approach of rewriting queries is based on approximate query mapping[5].

The approximate information filtering framework[6] has been proposed to deal with query in heterogeneous ontology-based information sources. The framework dealing with class and relation hierarchies and both the maximally and minimally contained reformulation has been presented in Ref. [7]. The framework is based on the assumption that one-to-one subsumption relations between ontologies are always known, which is overly optimistic. Moreover, it does not consider disjointed relations. This paper uses an association matrix to describe relations between concepts in two different ontologies, including one-to-one subsumption relations. Then an approximate querying process can be processed automatically. The disjoint relations between concepts are also considered. It leads to more accurate approximations than the methods currently in use.

1 Background

Concept-based information retrieval is the simplest and the most important part of ontology-based information systems[2]. In a concept-based information source, the pages, documents or any other items of information are classified according to an ontology. Then the source can answer users' queries expressed according to the ontology.

Definition 1 An information source is a set of information items. If every information item in an information source S has been classified into one or more concepts in an ontology O, i. e. the information items are individuals in O, then S is called a concept-based information source with respect to ontology O. Let the concepts in ontology O form the set $C = \{c_1, c_2, \ldots, c_n\}$. In each S, we introduce an interpretation

Received 2004-06-28.

Foundation items: The National Natural Science Foundation of China (No. 60373066, 60303024, 60403016), the National Basic Research Program of China (973 Program) (No. 2002CB312000), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20020286004).

Biographies: Kang Dazhou (1980—), male, graduate; Xu Baowen (corresponding author), male, doctor, professor, bwxu@seu.edu.cn.

function: c_i^{I} means the set of information items that belong to concept c_i.

Definition 2 A concept query is a Boolean expression on the set of concept names in an ontology, and is called a query for short in this paper. The queries in concept set C and their answers in information source S can be defined as

① Every concept name in C is a query, the answer to the query c_i is c_i^{I};

② If e is a query, then $\neg e$ is also a query, the answer to the query $\neg e$ is the information items in S that are not in e^{I};

③ If e_1 and e_2 are queries, then $e_1 \wedge e_2$, $e_1 \vee e_2$ are queries, the answers are $(e_1 \wedge e_2)^{\mathrm{I}} = e_1^{\mathrm{I}} \cap e_2^{\mathrm{I}}$, $(e_1 \vee e_2)^{\mathrm{I}} = e_1^{\mathrm{I}} \cup e_2^{\mathrm{I}}$.

There are heterogeneity problems when the information sources use different ontologies. Different ontologies do not have the same concept hierarchy. When query users and information sources use different ontologies, the query should be rewritten by replacing the concept names in the user's ontology by the concept names in the system's ontology[4]. In general, it is not possible to find exactly corresponding queries in a different ontology, but we can approximately rewrite using the subsumption relations between concepts in different ontologies[5].

In the approximate information filtering framework, the key problem is finding approximations of each concept[6]. Let S_1 and S_2 be two information sources using ontologies O_1 and O_2, respectively; let the set of concepts in O_1 be C_1 and the set of concepts in O_2 be C_2; let c, d be concepts. Here $c \subseteq d$ represents that c is a subclass of d, and $c \subset d$ means that c is a proper subclass of d, i. e. c and d are not equivalent. In any information source, $c \subseteq d \rightarrow c^{\mathrm{I}} \subseteq d^{\mathrm{I}}$. The direct superclasses and subclasses of a concept are useful for finding its approximations[7].

Definition 3 Let c be a concept in C_1, then the set of concepts $\mathrm{lub}(c, C_2)$ is called the least upper bounds of c in C_2, if for any concept $d \in C_2$ such that $c \subseteq d$ and there is no d' in C_2 such that $c \subseteq d' \subset d$, it is true that $d \in \mathrm{lub}(c, C_2)$.

Definition 4 Let c be a concept in C_1, then the set of concepts $\mathrm{glb}(c, C_2)$ is called the greatest lower bounds of c in C_2, if for any concept $d \in C_2$ such that $d \subseteq c$ and there is no d' in C_2 such that $d \subset d' \subseteq c$, it is true that $d \in \mathrm{glb}(c, C_2)$.

Using the least upper bounds and the greatest lower bounds, upper approximations and lower approximations of concept[6] can be defined as $\mathrm{ua}(c, C_2) = \bigwedge_{d_i \in \mathrm{lub}(c, C_2)} d_i$ and $\mathrm{la}(c, C_2) = \bigvee_{d_i \in \mathrm{glb}(c, C_2)} d_i$, respectively. Obviously, $\mathrm{la}(c, C_2) \subseteq c \subseteq \mathrm{ua}(c, C_2)$. Therefore, the upper and lower approximations can ensure the correctness or the completeness of the querying respectively, i. e. either the precise or recall of the results can be 100%[7].

The subsumption relations between concepts play important parts in the quality of approximate. How to find these relations between concepts in different ontologies automatically is an unsolved problem. Most methods leave this problem to the integration of ontologies, which is known as a very difficult task[8]. Learning methods can be used to automatically discover relations between concepts in different ontologies[9]. They use machine-learning methods based on the similarity measures between concepts to find mappings between concepts. The methods can only find one-to-one mappings; when handling no equivalent concepts and ambiguous concepts, it is impossible to determine an accurate match. It cannot find all the subclasses and the superclasses of a concept either. This paper uses an association matrix to automatically find all the subsumption relations.

Current methods in use only focus on the subsumption relations. The quality of the approximation may not be acceptable. In the worst case, the approximations always return an empty set or a full set as the result[10]. We consider the disjoint relations between concepts in our method to make the approximation more accurate.

2 Association Matrix

If there is a set of instances already categorized in both ontologies, we can generate an association matrix to learn the relations.

Definition 5 Let C_1 and C_2 be the sets of concepts of two ontologies O_1 and O_2. I is a set of instances which have been already categorized in both C_1 and C_2. The association matrix $\boldsymbol{M}$ of O_1 and O_2 is based on a mapping: $(C_1 \cup \{T\}) \times (C_2 \cup \{T\}) \rightarrow N$, where N is the set of non-negative integers, T is a top concept containing all the instances.

The size of $\boldsymbol{M}$ is $(|C_1| + 1) \times (|C_2| + 1)$ and it can be calculated as follows: let A be a concept in C_2, B be a concept in C_1. $\boldsymbol{M}[n(A)][n(B)]$ equals the number of instances belonging to both A and B; $\boldsymbol{M}[n(A)][n(T)]$ means the number of instances belonging to A; $\boldsymbol{M}[n(T)][n(B)]$ means the number of instances belonging to B; $\boldsymbol{M}[n(T)][n(T)]$ is the total number of instances; where $n(c)$ is the unique serial

number of each concept c: $n(c_1) \neq n(c_2)$ if c_1, c_2 are concepts in the same ontology, and for any c, $n(c) \neq n(T)$.

Some ontologies have a large set of concepts, which may make the matrix huge. We can comminute them because they often follow common scientific classifications. For example, most ontologies discriminate life-forms in animals, plants and other forms like microorganisms. Then concepts in animals and plants can be dealt with separately. We can build the association matrix for each domain.

Definition 6 Let $\boldsymbol{M}$ be the association matrix of C_1 and C_2. It has two normalized transform matrices: $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$. They satisfy:

$$\boldsymbol{M}_1[n(B)][n(A)] = \frac{\boldsymbol{M}[n(A)][n(B)]}{\boldsymbol{M}[n(T)][n(B)]}$$

$$\boldsymbol{M}_2[n(A)][n(B)] = \frac{\boldsymbol{M}[n(B)][n(A)]}{\boldsymbol{M}[n(A)][n(T)]}$$

where A is a concept in C_2 and B is a concept in C_1. $\boldsymbol{M}_1[n(B)][n(A)]$ represents the probability of the instances of B belonging to A. $\boldsymbol{M}_2[n(A)][n(B)]$ is the probability of the instances of A belonging to B.

The one-to-one subsumption relations between ontologies can be easily found from $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$:

If $\boldsymbol{M}_1[n(B)][n(A)] = 1$, then $B \subseteq A$

If $\boldsymbol{M}_2[n(A)][n(B)] = 1$, then $A \subseteq B$

3 Query Using the Association Matrix

With the association matrix, we can simply compute the upper and lower approximations of a concept by $\bigwedge_{\boldsymbol{M}_1[n(c)][n(d_i)]=1} d_i$ and $\bigvee_{\boldsymbol{M}_2[n(d_i)][n(c)]=1} d_i$. But there is a problem that not only the direct ones but all of the superclasses and subclasses are contained in the expression. It makes the approximation of a concept containing too many concepts and hard to be processed. This problem can be solved using the association matrix of the ontology and itself; let $\boldsymbol{M}'_1$ be the association matrix of C_1 and itself. From $\boldsymbol{M}'_1$, we can find all the superclasses and subclasses of a concept in C_1. We can get $\boldsymbol{M}'_2$ in the same way for C_2.

Here is a method to find the least upper bounds of c in C_2 using the association matrix: ① Find out all the concepts d_i in C_2 such that $\boldsymbol{M}_2[n(d_i)][n(c)] = 1$ and they form a set $\mathrm{ub}(c, C_2)$; ② For any d_i, d_j in $\mathrm{ub}(c, C_2)$ such that $i \neq j$, if $\boldsymbol{M}'_2[n(d_i)][n(d_j)] = 1$, then delete d_j from $\mathrm{ub}(c, C_2)$; ③ Do step ② until there is no d_i, d_j in $\mathrm{ub}(c, C_2)$ such that $i \neq j$ and $\boldsymbol{M}'_2[n(d_i)][n(d_j)] = 1$. Then we can get $\mathrm{lub}(c, C_2) = \mathrm{ub}(c, C_2)$; ④ If we get $\mathrm{lub}(c, C_2) = \varnothing$, let $\mathrm{lub}(c, C_2) = \{T\}$.

The greatest lower bounds can also be found: ① Find out all the concepts d_i in C_2 that $\boldsymbol{M}_2[n(d_i)][n(c)] = 1$ and they form a set $\mathrm{lb}(c, C_2)$; ② For any d_i, d_j in $\mathrm{lb}(c, C_2)$ such that $i \neq j$, if $\boldsymbol{M}'_2[n(d_i)][n(d_j)] = 1$, then delete d_j from $\mathrm{lb}(c, C_2)$; ③ Do step ② until there is no d_i, d_j in $\mathrm{lb}(c, C_2)$ such that $i \neq j$ and $\boldsymbol{M}'_2[n(c_i)][n(c_j)] = 1$. Then we can get $\mathrm{glb}(c, C_2) = \mathrm{lb}(c, C_2)$.

Except for removing some redundant equivalent members, the resulted $\mathrm{lub}(c, C_2)$ and $\mathrm{glb}(c, C_2)$ are the same as the least upper bounds and the greatest lower bounds defined in definitions 3 and 4. Then we can compute $\mathrm{la}(c, C_2)$ and $\mathrm{ua}(c, C_2)$. Let n be the number of concepts in each ontology; the computation complexity is no more than $O(n^3)$ to find approximations for all the concepts. This process only uses a simple matrix calculation and can be processed offline in advance.

The rewriting of a query requires that the original query be transformed into negation normal form, i. e. negations only apply to individual concept names but not to compound expressions. It can be done using the following two equations: $\neg(e_1 \wedge e_2) = \neg(e_1 \vee e_2)$ and $\neg(e_1 \vee e_2) = \neg(e_1 \wedge e_2)$.

If for every non-negated concept name c in the query, we replace c with $\mathrm{lub}(c, C_2)$; for every negated concept name c in C_1, we replace c with the $\mathrm{glb}(c, C_2)$. Then the answer to the new query on C_2 must also be the answer to the original query. The rewriting ensures the correctness.

If for every non-negated concept name c in C_1, we replace c with $\mathrm{glb}(c, C_2)$; for every negated concept name c in C_1, we replace c with $\mathrm{lub}(c, C_2)$. Then the answers to the new query on C_2 must contain all the answers to the original query. The rewriting ensures the completeness. Finally, the new queries on C_2 can be answered by the target system.

The method can be processed automatically, and is simple to implement. The computation complexity is linear to the size of the query expression.

4 Considering Disjoint Relation

We can make the problem simpler by using vectors to represent query expressions.

Definition 7 The concept vector of a query expression shows the probabilities of answers to this query belonging to each concept in an ontology (It does not have to be the same ontology that the query expression is based on). If an instance a is in the results of query e, and the concept vector of e is $\boldsymbol{v}$. And

① If $\boldsymbol{v}[n(c)]=1$, a must be an instance of concept c;

② If $\boldsymbol{v}[n(c)]=0$, a cannot be an instance of concept c;

③ If $0<\boldsymbol{v}[n(c)]<1$, a may be an instance of concept c.

If e is a concept c in C_1, we can define a vector

$$\boldsymbol{v}'(c, C_1)[i]=\begin{cases}1 & i=n(c)\\ 0 & i\neq n(c)\end{cases}$$

But it does not satisfy the definition when not all other concepts are disjoint with c. Especially, for any concept d that is superclass of c, if $\boldsymbol{v}$ is a concept vector of query c, it will be expected to have $\boldsymbol{v}[n(d)]=1$. We can calculate the concept vector of a concept name c in C_1: $\boldsymbol{v}(c, C_1)=\boldsymbol{v}'(c, C_1)\boldsymbol{M}_1'$, where $\boldsymbol{M}_1'$ is the normalized transform matrix of C_1 and the operator is defined as $(\boldsymbol{vM})[i]=\max_j(\boldsymbol{v}[j][j]\boldsymbol{M}[i])$. It is a concept vector of c in C_1.

The concept vector of c based on C_2 can be calculated by $\boldsymbol{v}(c, C_2)=\boldsymbol{v}'(c, C_1)\boldsymbol{M}_1$. If $\boldsymbol{v}$ is a concept vector of c based on C_i, d is a concept in C_j, it must be true that $c\subseteq d\Leftrightarrow\boldsymbol{v}(c, C_j)[n(d)]=1$. Similarly, $\boldsymbol{M}_1'\boldsymbol{v}'(c, C_1)^{\mathrm{T}}$, $\boldsymbol{M}_2\boldsymbol{v}'(c, C_1)^{\mathrm{T}}$ can show the subclasses of c in C_1 and C_2.

Considering disjoint relations will make the approximation more accurate. However, it may be that too many disjoint relations are real cases, and most of them are redundant. Using the concept vector, we propose a method to check which concepts need to be negated explicitly when rewriting c in C_2:

① Get the concept vector $\boldsymbol{v}$ of c in C_1 by $\boldsymbol{v}(c, C_1)=\boldsymbol{v}'(c, C_1)\boldsymbol{M}_1'$.

② Find the least upper bounds $\mathrm{lub}(c, C_2)$ of c in C_2.

③ Calculate a vector $\boldsymbol{v}_{\mathrm{lub}}=\boldsymbol{v}'_{\mathrm{lub}}\boldsymbol{M}_2'$, where

$$\boldsymbol{v}'_{\mathrm{lub}}[i]=\begin{cases}1 & \exists c(c\in\mathrm{lub}(c, C_2)\wedge i=n(c))\\ 0 & \text{otherwise}\end{cases}$$

④ Compare $\boldsymbol{v}(c, C_1)$ and $\boldsymbol{v}_{\mathrm{lub}}$, find all the concepts d in C_2 such that $\boldsymbol{v}(c, C_1)[n(d)]=0$, $\boldsymbol{v}_{\mathrm{lub}}[n(d)]>0$. Then form them into a new set named neg. $\boldsymbol{v}(c, C_1)[n(d)]=0$ means no answer is in concept d, $\boldsymbol{v}_{\mathrm{lub}}[n(d)]>0$ means the upper approximation is not implicated, so d needs to be negated explicitly.

⑤ Then simplify neg: For any d_i, $d_j\in$ neg, if $\boldsymbol{M}_2'[n(d_i)][n(d_j)]=1$, i. e. $c\subseteq\neg d_j\subseteq\neg d_i$, then delete d_i from neg; do this until there is no d_i, $d_j\in$ neg such that $\boldsymbol{M}'_2[n(d_i)][n(d_j)]=1$.

⑥ The new upper approximation of c in C_2 is

$$\mathrm{nua}(c, C_2)=\bigwedge_{d_i\in\mathrm{lub}(c, C_2)}d_i\wedge\bigwedge_{d_j\in\mathrm{neg}}\neg d_j$$

It is more accurate than the original upper approximation, since it has $c\subseteq\mathrm{nua}(c, C_2)\subseteq\mathrm{ua}(c, C_2)$. The recall of the new upper approximation is still 100%, but the precision is increased. Let n be the number of concepts in each ontology, the computation complexity is $O(n^3)$ to improve approximations of all the concepts. This process also only uses simple matrix calculation and can be processed offline in advance.

The lower approximation can be modified in a familiar way using $\boldsymbol{M}_2\boldsymbol{v}'(c, C_1)^{\mathrm{T}}$ and $\mathrm{glb}(c, C_2)$. The method needs to be improved in the future.

5 Conclusion

This paper proposes an approximate approach of querying between heterogeneous ontology-based information systems based on an association matrix. The association matrix and its normalized form can show the relations between concepts in ontologies. The association matrix is easy to generate.

We introduce the method that finds the approximations of concepts in another ontology using an association matrix. Then it rewrites the queries in one ontology to approximate queries in another ontology based on the subsumption relations between concepts in different ontologies. And the disjoint relations between concepts can be considered to make the approximation more accurate. It is simple to implement and expected to run quite fast.

The concept vector of the negation, conjunction and disjunction of concept vectors can be carefully defined. Then many query expressions can be simplified using association matrix before rewriting. This will make the query process more efficient. How the association matrix method can be used when facing multiple ontologies is for future work.

References

[1] Belkin N, Croft B. Information filtering and information retrieval: two sides of the same coin [J]. *Communications of the ACM*, 1992, **35**(12): 29 – 38.

[2] Shah U, Finin T, Joshi A, et al. Information retrieval on the semantic web [A]. In: *Proceedings of the* 10*th International Conference on Information and Knowledge Management*[C]. New York: ACM Press, 2003. 461 – 468.

[3] Guarino N. Formal ontology and information systems [A]. In: *Proceedings of the* 1*st International Conference on Formal Ontologies in Information Systems* [C]. Trento, Italy: IOS Press, 1998. 3 – 15.

[4] Chang K C-C, Garcia-Molina H. Mind your vocabulary:

query mapping across heterogeneous information sources [A]. In: *Proceedings of the ACM SIGMOD Conference* [C]. New York: ACM Press, 1999. 335 − 346.

[5] Chang K C-C, Garcia-Molina H. Approximate query mapping: accounting for translation closeness [J]. *The VLDB Journal*, 2001, **10**(2, 3): 155 − 181.

[6] Stuckenschmidt H. Approximate information filtering with multiple classification hierarchies [J]. *International Journal of Computational Intelligence and Applications*, 2002, **2**(3): 295 − 302.

[7] Akahani J, Hiramatsu K, Satoh T. Approximate query reformulation based on hierarchical ontology mapping [A]. In: *International Workshop on Semantic Web Foundations and Application Technologies* [C]. Nara, Japan, 2003. 43 − 46.

[8] Goasdoue F, Rousset M-C. Answering queries using views: a KRDB perspective for the semantic web [J]. *ACM Transactions on Internet Technology*, 2004, **4**(3): 255 − 288.

[9] Doan A, Madhavan J, Domingos P, et al. Learning to map between ontologies on the semantic web [A]. In: *Proceedings of the* 11*th International Conference on World Wide Web* [C]. Hawaii, USA, 2002. 662 − 673.

[10] Stuckenschmidt H. Ontology-based information sharing in weakly structured environments [D]. Amsterdam, Netherlands: AI Department of Vrije Universiteit Amsterdam, 2002.

基于关系矩阵的异构本体间近似查询

康达周[1]　徐宝文[1,2]　陆建江[1,2,3]　汪　鹏[1]

([1] 东南大学计算机科学与工程系, 南京 210096)

([2] 江苏省软件质量研究所, 南京 210096)

([3] 解放军理工大学理学院, 南京 210007)

摘要: 提出一种基于关系矩阵的异构本体信息系统间近似查询方法. 首先定义关系矩阵表达不同本体的概念间关系, 然后给出一个重写查询方法. 利用 2 个本体的概念间包含关系, 将一个本体中的查询重写为另一个本体中的近似查询, 从而解决本体异构问题. 并提出用向量表示查询, 通过查询向量和关系矩阵计算的结果考虑不同本体概念间的不相交关系, 得到比现有方法更精确的近似查询结果. 该方法可由机器自动完成, 易于实现且速度快.

关键词: 语义 web; 信息检索; 本体; 查询; 关系矩阵

中图分类号: TP311