Intelligent Query Answering

Zbigniew W. Ras*
Department of Computer Science
University of North Carolina
9201 University City Blvd.
Charlotte, NC 28223, USA
voice: +1 704-687-4567
fax: +1 704-687-3516
email: ras@uncc.edu

Agnieszka Dardzinska
Department of Mathematics
Bialystok Technical University
Wiejska 45a
15-351 Bialystok, Poland
voice: +48 85-746-9070
email: adardzin@uncc.edu

(*Corresponding author)

Intelligent Query Answering

Zbigniew W. Ras, University of North Carolina Charlotte, USA

Agnieszka Dardzinska, Bialystok Technical University, Poland

INTRODUCTION

One way to make Query Answering System (QAS) intelligent is to assume a hierarchical structure of its attributes. Such systems have been investigated by (Cuppens & Demolombe, 1988), (Gal & Minker, 1988), (Gaasterland et al., 1992) and they are called cooperative. Any attribute value listed in a query, submitted to cooperative QAS, is seen as a node of the tree representing that attribute. If QAS retrieves empty set of objects which match query q in a target information system S, then any attribute value listed in q can be generalized and the same the number of objects which possibly can match q in S can increase. In cooperative systems, these generalizations are usually controlled by users.

Another way to make QAS intelligent is to use knowledge discovery methods to increase the number of queries which QAS can answer: knowledge discovery module of QAS extracts rules from a local system S and requests their extraction from remote sites (if system is distributed). These rules are uses to construct new attributes and/or impute null or hidden values of attributes in S. By enlarging the set of attributes from which queries are built and by making information systems less incomplete, we not only increase the number of queries which QAS can handle but also increase the number of retrieved objects.

So, QAS based on knowledge discovery has two classical scenarios which need to be considered:

- In a standalone and incomplete system, association rules are extracted from that system and used to predict what values should replace null values before queries are answered.
- When system is distributed with autonomous sites and user needs to retrieve objects, from one of these sites (called client), satisfying query q based on attributes which are not local for that site, we search for definitions of these non-local attributes at remote sites and use them to approximate q (Ras, 2002), (Ras & Joshi, 1997), (Ras & Dardzinska, 2004).

The goal of this article is to provide foundations and basic results for knowledge-discovery based QAS.

BACKGROUND

Modern query answering systems area of research is related to enhancements of query-answering systems into intelligent systems. The emphasis is on problems in users posing queries and systems producing answers. This becomes more and more relevant as the amount of information available from local or distributed information sources increases. We need systems not only easy to use but also intelligent in answering the users' needs. A query-answering system often replaces human with expertise in the domain of interest, thus it is important, from the user's point of view, to compare the system and the human expert as alternative means for accessing information.

Knowledge systems are defined as information systems coupled with a knowledge base simplified in (Ras 2002), (Ras &Joshi, 1997), (Ras & Dardzinska, 1997) to a set of rules treated as definitions of attribute values. If information system is distributed with autonomous sites, these rules can be extracted either from the information system which is seen as local (query was submitted to that system) or from remote sites. Domains of attributes in the local information system *S* and the set of decision values used in rules from the knowledge base associated with *S* form the initial alphabet for the local query answering system. When the knowledge base associated with S is updated (new rules are added or some deleted), the alphabet for the local query answering system is automatically changed. In this paper we assume that knowledge bases for all sites are initially empty. Collaborative information system (Ras, 2002) learns rules describing values of incomplete attributes and attributes classified as foreign for its site called a client. These rules can be extracted at any site but their condition part should use, if possible, only terms which can be processed by the query answering system associated with the client. When the time progresses more and more rules can be added to the local knowledge base which means that some attribute values (decision parts of rules) foreign for the client are also added to its local alphabet. The choice of which site should be contacted first, in search for definitions of foreign attribute values, is mainly based on the number of attribute values common for the client and server sites. The solution to this problem is given in (Ras, 2002).

MAIN THRUST

The technology dimension will be explored to help clarify the meaning of intelligent query answering based on knowledge discovery and chase.

Intelligent Query Answering for Standalone Information System

QAS for an information system is concerned with identifying all objects in the system satisfying a given description. For example an information system might contain information about students in a class and classify them using four attributes of “hair color”, “eye color”, “gender” and “size”. A simple query might be to find all students with brown hair and blue eyes. When information system is incomplete, students having brown hair and unknown eye color can be handled by either including or excluding them from the answer to the query. In the first case we talk about optimistic approach to query evaluation while in the second case we talk about pessimistic approach. Another option to handle such a query would be to discover rules for eye color in terms of the attributes hair color, gender, and size. These rules could then be applied to students with unknown eye color to generate values that could be used in answering the query. Consider that in our example one of the generated rules said:

(hair, brown) $\wedge$ (size, medium) $\rightarrow$ (eye,brown).

Thus, if one of the students having brown hair and medium size has no value for eye color, then the query answering system should not include this student in the list of students with brown hair and blue eyes. Attributes hair color and size are classification attributes and eye color is the decision attribute.

We are also interested in how to use this strategy to build intelligent QAS for incomplete information systems. If query is submitted to information system S, the first step of QAS is to make S as complete as possible. The approach proposed in (Dardzinska & Ras, 2003b) is to use not only functional dependencies to chase S (Atzeni & DeAntonellis, 1992) but also use rules discovered from a complete subsystem of S to do the chasing.

In the first step, intelligent QAS identifies all incomplete attributes used in a query. An attribute is incomplete in S if there is an object in S with incomplete information on this attribute. The values of all incomplete attributes are treated as concepts to be learned (in a form of rules) from S.

Incomplete information in S is replaced by new data provided by Chase algorithm based on these rules. When the process of removing incomplete vales in the local information system is completed, QAS finds the answer to query in a usual way.

Intelligent Query Answering for Distributed Autonomous Information Systems

Semantic inconsistencies are due to different interpretations of attributes and their values among sites (for instance one site can interpret the concept “young” differently than other sites). Different interpretations are also due to the way each site is handling null values. Null value replacement by values suggested either by statistical or knowledge discovery methods is quite common before user query is processed by QAS.

Ontology (Guarino, 1998), (Sowa, 1999, 2000), (Van Heijst et al., 1997) is a set of terms of a particular information domain and the relationships among them. Currently, there is a great deal of interest in the development of ontologies to facilitate knowledge sharing among information systems.

Ontologies and inter-ontology relationships between them are created by experts in corresponding domain, but they can also represent a particular point of view of the global information system by describing customized domains. To allow intelligent query processing, it is often assumed that an information system is coupled with some ontology. Inter-ontology relationships can be seen as semantical bridges between ontologies built for each of the autonomous information systems so they can collaborate and understand each other.

In (Ras and Dardzinska, 2004), the notion of optimal rough semantics and the method of its construction have been proposed. Rough semantics can be used to model semantic inconsistencies among sites due to different interpretations of incomplete values of attributes. Distributed chase (Ras and Dardzinska, 2004) is a chase-type algorithm, driven by a client site of a distributed information system DIS, which is similar to chase algorithms based on knowledge discovery and presented in (Dardzinska and Ras, 2003a, 2003b). Distributed chase has one extra feature in comparison to other chase-type algorithms: the dynamic creation of knowledge bases at all sites of DIS involved in the process of solving a query submitted to the client site of DIS.

The knowledge base at the client site may contain rules extracted from the client information system and also rules extracted from information systems at remote sites in DIS. These rules are dynamically updated through the incomplete values replacement process (Ras and Dardzinska, 2004).

Although the names of attributes are often the same among sites, their semantics and granularity levels may differ from site to site. As the result of these differences, the knowledge bases at the client site and at remote sites have to satisfy certain properties in order to be applicable in a distributed chase.

So, assume that system S = (X,A,V), which is a part of DIS, is queried be user.

Chase algorithm, to be applicable to S, has to be based on rules from the knowledge base D associated with S which satisfies the following conditions:

1) Attribute value used in decision part of a rule from D has the granularity level either equal to or finer than the granularity level of the corresponding attribute in S.

2) The granularity level of any attribute used in the classification part of a rule from D is either equal or softer than the granularity level of the corresponding attribute in S.

3) Attribute used in the decision part of a rule from D either does not belong to A or is incomplete in S.

Assume again that S=(X,A,V) is an information system (Pawlak, 1991), (Ras and Dardzinska, 2004), where X is a set of objects, A is a set of attributes (seen as partial functions from X into $2^{(V\times[0,1])}$ and, V is a set of values of attributes from A. By [0,1] we mean the set of real numbers from 0 to 1. Let $L(D)=\{[t \rightarrow v_c] \in D: c \in In(A)\}$ be a set of all rules (called a knowledge-base) extracted initially from the information system S by ERID (Dardzinska and Ras, 2003c), where In(A) is a set of incomplete attributes in S.

Assume now that query q(B) is submitted to system S=(X,A,V), where B is the set of all attributes used in q(B) and that $A \cap B \neq \emptyset$. All attributes in B - $[A \cap B]$ are called foreign for S. If S is a part of a distributed information system, definitions of foreign attributes for S can be extracted at its remote sites (Ras, 2002). Clearly, all semantic inconsistencies and differences in granularity of attribute values among sites have to be resolved first. In (Ras and Dardzinska, 2004) only different granularity of attribute values and different semantics related to different interpretations of incomplete attribute values among sites have been considered.

In (Ras, 2002), it was shown that query q(B) can be processed at site S by discovering definitions of values of attributes from B - $[A \cap B]$ at the remote sites for S and next use them to answer q(B).

Foreign attributes for S in B, can be also seen as attributes entirely incomplete in S, which means values (either exact or partially incomplete) of such attributes should be ascribed by chase to all objects in S before query q(B) is answered. The question remains, if values discovered by chase are really correct?

Classical approach, to this kind of problems, is to build a simple DIS environment (mainly to avoid difficulties related to different granularity and different semantics of attributes at different sites). As the testing data set (Ras & Dardzinska, 2004a) have taken 10,000 tuples randomly selected from a database of some insurance company. This sample table, containing 100 attributes, was randomly partitioned into four subtables of equal size containing 2,500 tuples each. Next, from each of these subtables 40 attributes (columns) have been randomly removed leaving four data tables of the size 2,500×60 each. One of these tables was called a client and the remaining 3 have been called servers. Now, for all objects at the client site, values of one of the attributes, which was chosen randomly, have been hidden. This attribute is denoted by d. At each server site, if attribute d was listed in its domain schema, descriptions of d using See5 software (data are complete so it was not necessary to use ERID) have been learned. All these descriptions, in the form of rules, have been stored in the knowledge base of the client. Distributed Chase was applied to predict what is the real value of the hidden attribute for each object x at the client site. The threshold value $\lambda = 0.125$ was used to rule out all values predicted by distributed Chase with confidence below that threshold. Almost all hidden values (2476 out of 2500) have been discovered correctly (assuming $\lambda = 0.125$).

Distributed Chase and Security Problem of Hidden Attributes

Assume now that an information system $S=(X,A,V)$ is a part of DIS and attribute $b\in A$ has to be hidden. For that purpose, we construct $S_b=(X,A,V)$ to replace S, where:

1) $a_S(x) = a_{Sb}(x)$, for any $a \in A-\{b\}$, $x \in X$,

2) $b_{Sb}(x)$ is undefined, for any $x \in X$,

3) $b_S(x) \in V_b$.

Users are allowed to submit queries to S_b and not to S. What about the information system Chase(S_b)? How it differs from S?

If $b_S(x) = b_{Chase(Sb)}(x)$, where $x \in X$, then values of additional attributes for object x have to be hidden in S_b to guarantee that value $b_S(x)$ can not be reconstructed by Chase. In (Ras & Dardzinska, 2004a) it was shown how to identify the minimal number of such values.

FUTURE TRENDS

One of the main problems related to semantics of an incomplete information system S is the freedom how new values are constructed to replace incomplete values in S, before any rule extraction process begins. This replacement of incomplete attribute values in some of the slots in S can be done either by chase or/and by a number of available statistical methods (Giudici, 2003). This implies that semantics of queries submitted to S and driven (defined) by query answering system QAS based on chase may often differ. Although rough semantics can be used by QAS to handle this problem, we still have to look for new alternate methods.

Assuming different semantics of attributes among sites in DIS, the use of global ontology or local ontologies built jointly with inter-ontology relationships among them seems to be necessary for solving queries in DIS using knowledge discovery and chase. Still a lot of research has to be done in this area.

CONCLUSION

Assume that the client site in DIS is represented by partially incomplete information system S. When a query is submitted to S, its query answering system QAS will replace S by Chase(S) and next will solve the query using, for instance, the strategy proposed in (Ras and & Joshi, 1997). Rules used by Chase can be extracted from S or from its remote sites in DIS assuming that all differences in semantics of attributes and differences in granularity levels of attributes are resolved first. We can argue here why the resulting information system obtained by Chase can not be stored aside and reused when a new query is submitted to S? If system S is not frequently updated, we can do that by keeping a copy of Chase(S) and next reusing that copy when a new query is submitted to S. But, the original information system S still has to be kept so when user wants to enter new data to S, they can be stored in the original system. System

Chase(S), if stored aside, can not be reused by QAS when the number of updates in the original S exceeds a given threshold value. It means that the new updated information system S has to be chased again before any query is answered by QAS.

REFERENCES

Atzeni, P., DeAntonellis, V. (1992). *Relational Database Theory*, The Benjamin Cummings Publishing Company.

Cuppens, F., Demolombe, R. (1988). Cooperative answering: a methodology to provide intelligent access to databases, *Proceedings of the Second International Conference on Expert Database Systems*, 333-353.

Dardzinska, A., Ras, Z.W. (2003a). Rule-based Chase algorithm for partially incomplete information systems, *Proceedings of the Second International Workshop on Active Mining,* Maebashi City, Japan, October, 42-51.

Dardzinska, A., Ras, Z.W. (2003b). Chasing Unknown Values in Incomplete Information Systems, *Proceedings of ICDM'03 Workshop on Foundations and New Directions of Data Mining*, Melbourne, Florida, IEEE Computer Society, 24-30.

Dardzinska, A., Ras, Z.W. (2003c). On Rule Discovery from Incomplete Information Systems, *Proceedings of ICDM'03 Workshop on Foundations and New Directions of Data Mining,* Melbourne, Florida, IEEE Computer Society, 31-35.

Gal, A., Minker, J. (1988). Informative and cooperative answers in databases using integrity constraints, *Natural Language Understanding and Logic Programming*, North Holland, 277-300.

Gaasterland, T., Godfrey, P., Minker, J. (1992). Relaxation as a platform for cooperative answering, *Journal of Intelligent Information Systems 1 (3),* 293-321.

Giudici, P. (2003). *Applied Data Mining, Statistical Methods for Business and Industry*, Wiley, West Sussex, England.

Guarino, N., ed. (1998). *Formal Ontology in Information Systems*, IOS Press, Amsterdam.

Pawlak, Z., (1991). *Rough Sets-theoretical Aspects of Reasoning about Data*, Kluwer.

Ras, Z. (2002). Reducts-driven query answering for distributed knowledge systems, *International Journal of Intelligent Systems 17 (2)*, John Wiley & Sons, 113-124.

Ras, Z., Dardzinska, A. (2004). Ontology Based Distributed Autonomous Knowledge Systems, *Information Systems International Journal 29 (1),* Elsevier, 47-58.

Ras, Z., Dardzinska, A. (2004a). Data Security and Null Value Imputation in Distributed Information Systems, in *Advances in Soft Computing*, Proceedings of MSRAS'04 Symposium, Poland, Springer-Verlag, 2004, will appear.

Ras, Z., Joshi, S. (1997). Query approximate answering system for an incomplete DKBS, *Fundamenta Informaticae 30 (3),* IOS Press, 313-324.

Sowa, J.F. (1999). Ontological categories, in L. Albertazzi, ed., *Shapes of Forms: From Gestalt Psychology and Phenomenology to Ontology and Mathematics*, Kluwer, 307-340.

Sowa, J.F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole Publishing Co., Pacific Grove, CA.

Van Heijst, G., Schreiber, A., Wielinga, B. (1997). Using explicit ontologies in KBS development, *International Journal of Human and Computer Studies 46, (2/3),* 183-292.

TERMS AND THEIR DEFINITION

Autonomous information system: Information system existing as an independent entity.

Chase: Kind of a recursive strategy applied to a database V, based on functional dependencies or rules extracted from V, by which a null value or an incomplete value in V is replaced by a new more complete value.

Distributed chase: Kind of a recursive strategy applied to a database V, based on functional dependencies or rules extracted both from V and other autonomous databases, by which a null value or an incomplete value in V is replaced by a new more complete value. Any differences in semantics among attributes in the involved databases have to be resolved first.

Intelligent query answering: Enhancements of query-answering systems into sort of intelligent systems (capable or being adapted or molded). Such systems should be able to interpret incorrectly posed questions and compose an answer not necessarily reflecting precisely what is directly referred to by the question, but rather reflecting what the intermediary understands to be the intention linked with the question.

Knowledge base: A collection of rules defined as expressions written in predicate calculus. These rules have a form of associations between conjuncts of values of attributes.

Ontology: An explicit formal specification of how to represent objects, concepts and other entities that are assumed to exist in some area of interest and relationships holding among them. Systems that share the same ontology are able to communicate about domain of discourse without necessarily operating on a globally shared theory. System commits to ontology if its observable actions are consistent with the definitions in the ontology.

Query semantics: The meaning of a query with an information system as its domain of interpretation. Application of knowledge discovery and Chase in query evaluation makes semantics operational.

Semantics: The meaning of expressions written in some language, as opposed to their syntax which describes how symbols may be combined independently of their meaning.