Cooperative multi-hierarchical query answering systems

Zbigniew W. Ras[1,2], Agnieszka Dardzinska[3]
[1] *Univ. of North Carolina, Dept. of Computer Science, Charlotte, N.C. 28223, USA*
[2] *Polish Academy of Sciences, Institute of Comp. Science, 01-237 Warsaw, Poland*
[3] *Bialystok Technical Univ., Dept. of Computer Science, 15-351 Bialystok, Poland*

Article Outline

Glossary
I. Definition of the Subject and Its Importance
II. Introduction
III. Multi-hierarchical Decision System
IV. Cooperative Query Answering
V. Future Directions
VI. Bibliography

Glossary

Autonomous information system
Autonomous information system is an information system existing as an independent entity.

Intelligent query answering
Intelligent query answering is an enhancements of query-answering into sort of an intelligent system (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
Knowledge base is 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
Ontology is 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.

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.

I. Definition of the Subject and Its Importance.

One way to make Query Answering System (QAS) intelligent is to assume the hierarchical structure of their attributes. Such systems have been investigated by (Cuppens & Demolombe, 1988), (Gal & Minker, 1988), (Gaasterland et al., 1992) and they are called cooperative. Queries submitted to them are built, in a classical way, from values of attributes describing objects in an information system S and from two-argument functors "and", "or". Instead of "or", we use symbol "+". Instead of "and", we use symbol "*". Let us assume that QAS is associated with an information system S. Now, if query q submitted to QAS fails, then any attribute value listed in q can be generalized and the same the number of objects supporting q in S may increase. In cooperative systems, these generalizations are controlled either by users (Gal & Minker, 1988), or by methods based on knowledge discovery (Muslea, 2004). Conceptually, a similar approach has been proposed by (Lin, 1989). He defines a neighborhood of an attribute value which we can interpret as its generalization (or its parent in the corresponding hierarchical attribute structure). When query fails, then the query answering system is trying to replace values in a query by new values from their corresponding neighborhoods. QAS for S can also collaborate and exchange knowledge with other information systems. In all such cases, it is called intelligent. In papers (Ras & Dardzinska, 2004, 2006 query answering strategy was based on a guided process of knowledge (rules) extraction and knowledge exchange among systems. Knowledge extracted from information systems collaborating with S was used to construct new attributes in S and/or impute null or hidden values of attributes in S. This way we do not only enlarge the set of queries which QAS can successfully answer but also increase the overall number of retrieved objects and their confidence. Some attributes in S can be distinguished. We usually call them decision attributes. Their values represent concepts which can be defined in terms of the remaining attributes in S, called classification attributes. Query languages for such information systems are built only from values of decision attributes and from two-argument functors "+", "*" (Ras et al., 2007). The semantics of queries is defined in terms of semantics of values of classification attributes. Precision and recall of QAS is strictly dependent on the support and confidence of the classifiers used to define queries.

II. Introduction.

Responses by QAS to submitted queries do not always contain the information desired and although they may be logically correct, can sometimes be misleading. Research in the area of intelligent query answering rectifies these problems. Classical approach is based on cooperative method called relaxation for expanding an information system and related to it queries (Cuppens & Demolombe, 1988), (Gal & Minker, 1988). The relaxation method expands the scope of a query by relaxing the constraints implicit in the query. This allows QAS to return answers related to the original query as well as the literal answers which may be of interest to the user.

This paper concentrates on multi-hierarchical decision systems which are defined as information systems with several hierarchical distinguished attributes called decision attributes. Their values are used to build queries. We give the theoretical framework for modeling such systems and its corresponding query languages. Standard interpretation and the classifier-based interpretation of queries are introduced and used to model the quality (precision, recall) of QAS.

III. Multi-hierarchical Decision System

In this section we introduce the notion of a multi-hierarchical decision system S and the query language built from atomic expressions containing only values of the decision attributes in S. Classifier-based semantics and the standard semantics of queries in S are proposed. The set of objects X in S is defined as the interpretation domain for both semantics. Standard semantics identifies all objects in X which should be retrieved by a query. Classifier-based semantics gives weighted sets of objects which are retrieved by queries. The notion of precision and recall of QAS in the proposed setting is introduced. We use only rule-based classifiers to define the classifier-based semantics. By improving the confidence and support of the classifiers we improve the precision and recall of QAS.

Definition 1

By a multi-hierarchical decision system we mean a triple $S = (X, A \cup D, V)$, where X is a nonempty, finite set of objects, $D = \{d[i]: 1 \leq i \leq k\}$ is a set of hierarchical decision attributes, A is a nonempty finite set of classification attributes, and $V = \bigcup\{V_a : a \in A \cup D\}$ is a set of their values.

We assume that:

- V_a, V_b are disjoint for any $a, b \in A \cup D$, such that $a \neq b$,

  $a : X \rightarrow V_a$ is a partial function for every $a \in A \cup D$.

Definition 2

By a set of decision queries (d-queries) for S we mean a least set T_D such that:

- $0, 1 \in T_D$,
- if $w \in \bigcup\{V_a : a \in D\}$, then $w, \sim w \in T_D$,
- if $t_1, t_2 \in T_D$, then $(t_1 + t_2), (t_1 * t_2) \in T_D$.

Definition 3

Decision query t is called simple if $t = t_1 * t_2 * \ldots * t_n$ and

$(\forall j \in \{1,2,\ldots,n\})[(t_j \in \bigcup\{V_a : a \in D\}) \vee (t_j = \sim w \wedge w \in \bigcup\{V_a : a \in D\})]$.

Definition 4

By a set of classification terms (c-terms) for S we mean a least set T_C such that:

- $0, 1 \in T_C$,
- if $w \in \bigcup\{V_a : a \in A\}$, then $w, \sim w \in T_C$,
- if $t_1, t_2 \in T_C$, then $(t_1 + t_2), (t_1 * t_2) \in T_C$.

Definition 5

Classification term t is called simple if $t = t_1 * t_2 * \ldots * t_n$ and

$(\forall j \in \{1,2,\ldots,n\})[(t_j \in \bigcup\{V_a : a \in A\}) \vee (t_j = \sim w \wedge w \in \bigcup\{V_a : a \in A\})]$.

Definition 6

By a classification rule we mean any expression of the form $[t_1 \rightarrow t_2]$, where t_1 is a simple classification term and t_2 is a simple decision query.

Definition 7

Semantics M_S of c-terms in $S = (X, A \cup D, V)$ is defined in a standard way as follows:

- $M_S(0) = 0$, $M_S(1) = X$,
- $M_S(w) = \{x \in X : w = a(x)\}$ for any $w \in V_a$, $a \in A$,
- $M_S(\sim w) = \{x \in X : (\exists v \in V_a)[v = a(x) \ \& \ v \neq w]\}$ for any $w \in V_a$, $a \in A$,
- if t_1, t_2 are terms, then

  $M_S(t_1 + t_2) = M_S(t_1) \cup M_S(t_2)$,

  $M_S(t_1 * t_2) = M_S(t_1) \cap M_S(t_2)$.

Now, we introduce the notation used for values of decision attributes. Assume that the term d[i] also denotes the first granularity level of a hierarchical decision attribute d[i]. The set {d[i,1], d[i,2], d[i,3],…} represents the values of attribute d[i] at its second granularity level. The set $\{d[i,1,1], d[i,1,2],\ldots, d[i,1,n_i]\}$ represents the values of attribute d at its third granularity level, right below the node d[i,1]. We assume here that the value d[i,1] can be refined to any value from $\{d[i,1,1], d[i,1,2],\ldots,d[i,1,n_i]\}$, if necessary. Similarly, the set {d[i,3,1,3,1], d[i,3,1,3,2], d[i,3,1,3,3], d[i,3,1,3,4]} represents the values of attribute d at its forth granularity level which are finer than the value d[i,3,1,3].

Now, let us assume that a rule-based classifier is used to extract rules describing simple decision queries in *S*. We denote that classifier by **RC**. The definition of semantics N_S of c-terms is **RC** independent whereas the definition of semantics M_S of d-queries is **RC** dependent.

Definition 8

Classifier-based semantics M_S of d-queries in $S = (X, A \cup D, V)$ is defined as follows:

- if t is a simple d-query in S and $\{r_j = [t_j \rightarrow t]: j \in J_t\}$ is a set of all rules defining t which are extracted from S by classifier **RC**, then

  $M_S(t) = \{(x,p_x): (\exists j \in J_t)(x \in M_S(t_j)[p_x =$

  $\Sigma\{conf(j)\cdot sup(j): x \in M_S(t_j) \ \& \ j \in J_t\}/\Sigma\{sup(j): x \in M_S(t_j) \ \& \ j \in J_t\}]\}$, where conf(j), sup(j) denote the confidence and the support of the rule $[t_j \rightarrow t]$, correspondingly.

Definition 9

Attribute value $d[j_1, j_2,\ldots j_n]$ in $S = (X, A \cup D, V)$ is dependent on $d[i_1, i_2,\ldots, i_k]$ in S, if one of the following conditions hold:

1) $n \leq k \ \& \ (\forall m \leq n)[i_m = j_m]$,
2) $n > k \ \& \ (\forall m \leq k)[i_m = j_m]$.

Otherwise, $d[j_1, j_2,\ldots j_n]$ is called independent from $d[i_1, i_2,\ldots, i_k]$ in S.

Example 1

The attribute value d[**2,3,1,2**] is dependent on the attribute value d[**2,3,1,2**,5,3]. Also, d[**2,3,1,2,5,3**,2,4] is dependent on d[**2,3,1,2,5,3**].

Definition 10

Let $S = (X, A \cup \{d[1],d[2],..,d[k]\}, V)$, $w \in V_{d[i]}$, and $IV_{d[i]}$ be the set of all attribute values in $V_{d[i]}$ which are independent from w.

Standard semantics N_S of d-queries in S is defined as follows:

- $N_S(0) = 0$, $N_S(1) = X$,
- if $w \in V_{d[i]}$, then $N_S(w) = \{x \in X : d[i](x)=w\}$, for any $1 \leq i \leq k$
- if $w \in V_{d[i]}$, then $N_S(\sim w) = \{x \in X : (\exists v \in IV_{d[i]})[d[i](x)=v]\}$, for any $1 \leq i \leq k$
- if t_1, t_2 are terms, then

$N_S(t_1 + t_2) = N_S(t_1) \cup N_S(t_2)$,

$N_S(t_1 * t_2) = N_S(t_1) \cap N_S(t_2)$.

Definition 11

Let $S = (X, A \cup D, V)$, t is a d-query in S, $N_S(t)$ is its meaning under standard semantics, and $M_S(t)$ is its meaning under classifier-based semantics. Assume that $N_S(t) = X_1 \cup Y_1$, where $X_1 = \{x_i, i \in I_1\}$, $Y_1 = \{y_i , i \in I_2\}$. Assume also that $M_S(t) = \{(x_i, p_i): i \in I_1\} \cup \{(z_i, q_i): i \in I_3\}$ and $\{y_i , i \in I_2\} \cap \{z_i , i \in I_3\} = \emptyset$.

By precision of a classifier-based semantics M_S on a d-query t, we mean

$$rec(M_S, t) = [\Sigma\{p_i : i \in I_1\} + \Sigma\{(1 - q_i) : i \in I_3\}]/[card(I_1) + card(I_3)].$$

By recall of a classifier-based semantics M_S on a d-query t, we mean

$$Rec(M_S, t) = [\Sigma\{p_i : i \in I_1\}]/[card(I_1) + card(I_3)].$$

Example 2

Assume that $N_S(t) = \{x_1, x_2, x_3, x_4\}$, $M_S(t) = \{(x_1, p_1), (x_2, p_2), (x_5, p_5), (x_6, p_6)\}$.
Then:

$Prec(M_S, t) = [p_1 + p_2 + (1-p_5) + (1 - p_6)]/4$,

$Rec(M_S, t) = [p_1 + p_2]/4$.

IV. Cooperative Query Answering

There are cases when classical Query Answering Systems (QAS) fail to return any answer to a d-query q but still a satisfactory answer can be found. For instance, let us assume that in a multi-hierarchical decision system $S = (X, A \cup D, V)$, where $D = \{d[1],d[2],..,d[k]\}$, there is no single object which description matches the query q. Assuming that a distance measure between objects in S is defined, then by generalizing q, we may identify objects in S which descriptions are closest to the description of q. This problem is similar to the problem when the granularity of an attribute value used in a query q is finer than the granularity of the corresponding attribute used in S. By replacing such attribute values in q by more general values used in S, we may retrieve objects from S which satisfy q.

Definition 12

The distance δ_S between two attribute values $d[j_1, j_2,\ldots j_n]$, $d[i_1, i_2,\ldots, i_m]$ in $S = (X, A \cup D, V)$, where $j_1 = i_1$, $p \geq 1$, is defined as follows:

1) if $[j_1, j_2,\ldots j_p] = [i_1, i_2,\ldots, i_p]$ and $j_{p+1} \neq i_{p+1}$, then $\delta_S[d[j_1, j_2,\ldots j_n], d[i_1, i_2,\ldots, i_m]] = 1/[2^{p-1}]$
2) if $n \leq m$ and $[j_1, j_2,\ldots j_n] = [i_1, i_2,\ldots, i_n]$, then $\delta_S[d[j_1, j_2,\ldots j_n], d[i_1, i_2,\ldots, i_m]] = 1/[2^n]$

The second condition, in the above definition, represents the average case between the best and the worth case.

Example 3

Following the above definition of the distance measure, we get:

1) $\delta_S[d[2,3,2,4], d[2,3,2,5,1]] = ¼$
2) $\delta_S[d[2,3,2,4], d[2,3,2]] = 1/8$

Let us assume that q = q(a[3,1,3,2], b[1], c[2]) is a d-query submitted to S. The notation q(a[3,1,3,2], b[1], c[2]) means that q is built from a[3,1,3,2], b[1], c[2] which are the atomic attribute values in S. Additionally, we assume that attribute a is not only hierarchical but also it is ordered. It basically means that the difference between the values a[3,1,3,2] and a[3,1,3,3] is smaller than between the values a[3,1,3,2] and a[3,1,3,4]. Also, the difference between any two elements in {a[3,1,3,1], a[3,1,3,2], a[3,1,3,3], a[3,1,3,4]} is smaller than between a[3,1,3] and a[3,1,2].

Now, we outline a possible strategy which QAS can follow to solve q. Clearly, the best solution for answering q is to identify objects in S which precisely match the d-query submitted by user. If it fails, we try to identify objects which match d-query q(a[3,1,3], b[1], c[2]). If we succeed, then we try d-queries q(a[3,1,3,1], b[1], c[2]) and q(a[3,1,3,3], b[1], c[2]). If we fail, then we should succeed with q(a[3,1,3,4], b[1], c[2]). If we fail with q(a[3,1,3], b[1], c[2]), then we try q(a[3,1], b[1], c[2]) and so on.

To present this cooperative strategy in a more precise way, we use an example and start with a very simple dataset. Namely, we assume that S has 4 decision attributes which belong to the set {a, b, c, d}. System S contains only four objects listed below

X	e	f	g	.....	.....	a	b	c	d
x_1	e[1]	f[1]	.....	.....	.....	a[1]	b[2]	c[1,1]	d[3]
x_2	e[2]	f[1]	.....	.....	.....	a[1,1]	b[2,1]	c[1,1,1]	d[3,1,2]
x_3	e[2]	f[1]	.....	.....	.....	a[1,1,1]	b[2,2,1]	c[2,2]	d[1]
x_4	e[1]	f[2]	.....	.....	.....	a[2]	b[2,2]	c[1,1]	d[1,1]

Table 1. Multi-hierarchical decision system S

Now, we assume that d-query q = a[1,2]∗b[2]∗c[1,1]∗d[3,1,1] is submitted to the multi-hierarchical decision system S (see Table 1). Clearly, q fails in S.

Jointly with q, also a threshold value for a minimum support can be supplied as a part of a d-query. This threshold gives the minimal number of objects that need to be returned as an answer to q. When QAS fails to answer q, the nearest objects satisfying q have to be identified.

The algorithm for finding these objects is based on the following steps:

If QAS fails to identify sufficient number of objects satisfying q in S, then the generalization process starts. We can generalize either attribute a or d. Since the value d[3,1,2] has lower granularity level than a[1,1], then we generalize d[3,1,2] getting a new query q_1 = a[1,2]∗b[2]∗c[1,1] ∗d[3,1]. But q_1 still fails in S. Now, we generalize a[1,1] getting a new query q_2 = a[1]∗b[2]∗c[1,1] ∗d[3,1]. Objects x_1, x_2 are the only objects in S which support q_2.

If the user is only interested in one object satisfying the query q, then we need to identify which object in $\{x_1, x_2\}$ has a distance closer to q.

Clearly,
$\delta_S[q, x_1] = \delta_S[[a[1,2], b[2], c[1,1], d[3,1,1]], [a[1], b[2],c[1,1], d[3]]] =$
¼+0+0+1/4=1/2,
$\delta_S[q, x_2] = \delta_S[[a[1,2], b[2], c[1,1], d[3,1,1]], [a[1,1],b[2,1],c[1,1,1],d[3,1,2]]] =$
1/4+1/4+1/8+1/8 = ¾, which means x_1 is the winning object.

Let us notice that the cooperative strategy only identifies objects satisfying d-queries and the same it identifies objects to be returned by QAS to the user. The confidence assigned to these objects depends on the classifier **RC**.

V. Future Directions

We have introduced the notion of a system-based semantics and user-based semantics of queries. User-based semantics is associated with the indexing of objects done by a user which is time consuming and unrealistic for very large sets of data. System-based semantics is associated with automatic indexing of objects in X which strictly depends on the support and confidence of classifiers and depends on the precision and recall of a query answering system. The quality of classifiers can be improved by a proper enlargement of the set X and the set of describing them features which differentiate the real-life objects from the same semantic domain as X in a better way. An example, for instance, is given in (Ras et al, 2007). The quality of a query answering system (QAS) can be improved by its cooperativeness. Both precision and recall of QAS is getting increased if no-answer queries are replaced by generalized queries which are answered by QAS on a higher granularity level than the initial level of queries submitted by users. Assuming that system is distributed, the quality of QAS for multi-hierarchical decision system S can be also improved through collaboration among sites (Ras, Z. & Dardzinska, A., 2004, 2006).

The key concept of intelligent QAS based on collaboration among sites is to generate global knowledge through knowledge sharing. Each site develops knowledge independently which is

used jointly to produce global knowledge. Assume that two sites S_1 and S_2 accept the same ontology of their attributes and share their knowledge in order to solve a user query successfully. Also, assume that one of the attributes at site S_1 is confidential. The confidential data in S_1 can be hidden by replacing them with null values. However, users at S_1 may treat them as missing data and reconstruct them with the knowledge extracted from S_2 (Im, S., Ras, Z., 2007). The vulnerability illustrated in this example shows that a security-aware data management is an essential component for any intelligent QAS to ensure data confidentiality.

VI. Bibliography

Primary Literature

1. Chmielewski, M.R., Grzymala-Busse, J.W., Peterson, N.W. (1993) The rule induction system LERS - a version for personal computers, Foundations of Computing and Decision Sciences, Vol. 18, No. 3-4, Institute of Computing Science, Technical University of Poznan, Poland, pp 181-212
2. Chu, W., Yang, H., Chiang, K., Minock, M., Chow, G., Larson, C. (1996) Cobase: A scalable and extensible cooperative information system, Journal of Intelligent Information Systems, Vol. 6, No. 2/3, pp 223-259
3. Gaasterland, T. (1997) Cooperative answering through controlled query relaxation, IEEE Expert, Vol. 12, No. 5, pp 48-59
4. Giannotti, F., Manco, G. (2002) Integrating data mining with intelligent query answering, Logics in Artificial Intelligence, LNCS 2424, pp 517-520
5. Godfrey, P. (1993) Minimization in cooperative response to failing database queries, in International Journal of Cooperative Information Systems, Vol. 6, No. 2, pp 95-149
6. Guarino, N., ed. (1998) Formal ontology in information systems, IOS Press, Amsterdam
7. Im, S., Ras, Z. (2007) Protection of sensitive data based on reducts in a distributed knowledge discovery system, Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE 2007), in Seoul, South Korea, IEEE Computer Society, pp 762-766
8. Lin, T.Y. (1989) Neighborhood systems and approximation in relational databases and knowledge bases, Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems, Poster Session Program, Oak Ridge National Laboratory, ORNL/DSRD-24, pp 75-86
9. Muslea, I. (2004) Machine Learning for Online Query Relaxation, Proceedings of KDD-2004, in Seattle, Washington, ACM, pp 246-255
10. Pawlak, Z. (1981) Information systems - theoretical foundations, Information Systems Journal, Vol. 6, pp 205-218
11. Ras, Z.W., Dardzinska, A. (2006) Solving Failing Queries through Cooperation and Collaboration, Special Issue on Web Resources Access, (Editor: M.-S. Hacid), World Wide Web Journal, Springer, Vol. 9, No. 2, pp 173-186
12. Ras, Z., Dardzinska, A. (2004) Ontology based distributed autonomous knowledge systems, Information Systems International Journal 29 (1), Elsevier, pp 47-58
13. Ras, Z.W., Zhang, X., Lewis, R. (2007) MIRAI: Multi-hierarchical, FS-tree based Music Information Retrieval System, (Invited Paper), Proceedings of RSEISP 2007, M. Kryszkiewicz et al. (Eds), LNAI, Vol. 4585, Springer, pp 80-89