MODELLING THE MAN MACHINE INTERACTION
Erotetic Logic and Information Retrieval Systems

Antonio Bellacicco
Department of Theory of Systems and Organizations, University of Teramo, Viale Crucioli,122, Teramo, Italy

Mario Vacca
"B. Pascal" High School, Via P.Nenni, 48, Pomezia, Italy

Keywords: Question, answer, search engine, erotetic logic

Abstract: The usual communication between man and machine is a one way interaction. It can be upgraded considering a two way interaction if the basic constituents of an information retrieval system are deeply modified in their principles. In this paper we redefine, in a syntactical way, some concepts of the erotetic logic to make them more easily computable and show how them can be used to solve some problems in the field of information retrieval systems. The result is the possibility to build more flexible and powerful information retrieval systems.

1 INTRODUCTION

Cognitive science, data mining, information retrieval or artificial intelligence, among other areas, analyze how information are requested and exchanged by two dialogists; the view proposed in this paper mainly refers to the last two mentioned areas.

In retrieval systems, the questioner communicates with the questionee by a query language and the questionee is usually modelled by an engine able to search a database or a knowledge base (from now KB for short) for the information requested by the questioner. Most difficulties in the use of search engines are just related to the narrowness of the query language used (for example, it is tied to isolated words and not to an entire meaningful sentences) and to the impossibility of interacting. Furthermore, the inefficiency of the search or the absence of any appropriate strategies when too many answers or no one are found often characterize search engines. These problems, among others, lead the research to different directions in order to make more versatile and powerful the information retrieval or the Q/A systems, as pointed out by many authors among which Bellacicco (2002, 2003), Burhans (2002) or Loia et al. (2002).

In this paper we firstly redefine some concepts of the erotetic logic in a syntactic way and hence we show how these concepts can be exploited to solve the problems of "multiple answers" and "plausible answers". As a consequence, these concepts are seen as the core for a search engine able to output either a sentence or another question.

Because of the paper is not self-contained, we refer to Groenendijk and Stochow (1997) for an introduction about the theory of questions and references or to Ginzburg (1995a, b), Krifka (2001), Piwek (1997), Ram (1991), Wisniewki (1995) for different approaches.

After a brief illustration of the basic functionality foreseen for the model, the third section is entirely devoted to questions: firstly a sketch of the logical representation used for them, next some transformation rules are discussed and examples of their use are reported. Finally, in the fourth part, the jump of domain is presented: a way to answer to questions using the similarity among concepts.

In the sequel, we use the terms query, question, response and answer with the following meaning:
query: a question in a natural language form;
question: pure logical form of a query;
response: answer to a query;
answer: logical form of the response to a question.
We use the term answer when confusion do not occurs.

Furthermore, when no ambiguities occur, we also denote by $\underline{x}$ the n-tuple $x_1,\ldots,x_n$. IRS stands for

Bellacicco A. and Vacca M. (2004).
MODELLING THE MAN MACHINE INTERACTION - Erotetic Logic and Information Retrieval Systems.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 172-179
DOI: 10.5220/0001139501720179

Information Retrieval System.

2 THE MODEL: FOUNDATIONS AND FEATURES

The factors that usually characterize information retrieval systems are:

1) the organization of the knowledge base;
2) the features of the query language;
3) the concept of answer used (answerhood);
4) the search algorithms.

As above mentioned, many attempts to improve the performances through the extension of one or more of these features had been made. Obviously, because of the strong connections, it is not possible to extend significantly one of the previous mentioned features without modifying the remaining ones. We started the analysis from the well known consideration that the memory organization is crucial to the performances of an IRS. It had been indeed showed by many authors, like Schank (1982), Cautiero et al. (1991), Kolodner (1983a, b) and Lebowitz (1988), that an efficient storage of information leads either to forgetting or to a powerful ability in reconstructing episodes. In this paper we are concerned with retrieval methods that don't explore the entire Knowledge Base in order to find some information. Following Schank (1982), KB is constituted by concepts organized by means of two kinds of links: generalization and packaging. Therefore the search algorithm will know only the structure (the organization principles) of KB.

As we suppose such a memory organization, it follows that a basic mechanism to search needs a more dynamic and flexible algorithm than the existing one.

Erotetic reasoning seems adequate to perform this task. The importance of the erotetic reasoning had been pointed out by many authors: Schank (1986) showed that questions play a crucial role in understanding and that reasoning with questions helps to find or build up answers (also creatively); Wisniewsky (1994, 1995) introduced the semantic notion of the evocation (the generation of questions from declarative sentences) and of the erotetic implication (the generation of questions from questions and declarative sentences). Unfortunately, both approaches are very difficult to apply: the first one because it gives specific domain dependent rules primarily, while the second one, yet being very general, considers the logic of questions from the semantics point of view, with the consequent processing difficulties.

In this paper, mainly concerned with the extension of the search engine, we propose some general syntactic rules to transform questions in order to search a base of knowledge in an "intelligent" way. Even thought it is inspired both to the cognitive and erotetic logic approaches, our proposal is different from those in which the rules to transform questions are of a syntactic nature. The main ideas are synthesized in the following two principles:

1) answers to a given question are defined in a syntactic way;
2) questions are transformed syntactically in order to find suitable answers.

We foresee to embody the following features in the model we propose:

1)communication abilities, to rule the flow query/answer between questioner and questionee;
2)search engine/question processor, that takes a question to a declarative sentence or to another question;
3)parser, that inputs a query (or a declarative sentence) in natural language form and outputs its logical representation; the translation process we think of is a two stage one:
query→ (intermediate form)→ question (logical form);
4)output, that translates a logical sentence in a natural language.

3 THE ANSWER RETRIEVAL

We follow the non reductionistic approach, in which questions are specific expressions not reducible to any other object. Therefore, we will consider a logic language with both an assertoric and an erotetic part, the latter consisting of various kinds of questions.

3.1 Question representation and answerhood

The representation we will use is like that proposed by Wisniewski (1995).

Broadly speaking, a logical question consists of five parts: the erotetic symbol ? (telling that the formula is a question), a list of variables (what the question asks for), a list of categorial qualifiers (that specifies the categories of object involved in the question), an erotetic symbol (that specifies the kind of question) and a body (a predicate that represent the core of the question), in agreement with the following syntax:

?<list of the variables> <list of the categorial qualifiers> < symbol specifying the kind> <body>

A question will be structured as follow:

$?x_{k1}..x_{km}[P_1(x_1,\ldots,x_n)]...[P_m(x_1,\ldots,x_n)]\Sigma\{B(x_1,\ldots,x_n)\}$
where:
-? indicates an interrogative formula;
-$x_{k1}..x_{km}$ are variables;
-predicates P_i are categorial qualifiers;
-$\Sigma \in \{S, W, O, U, T\}$ or is absent
-the predicate $B(x_1,\ldots,x_n)$ is the body of the question.
In the case that both Σ and list of variables are absent, we have a yes-no question.
See Wisniewski (1995) for a detailed account about the meaning of the previous constants.

For example, the query "Have Begin and Vance wives ever met?" could be translated as:
$?\lambda t[time(t)][woman(\alpha)][woman(\beta)][wife(\alpha,Vance)]$
$[wife(\beta,Begin)]S\{occurs(t,meet(\alpha,\beta))\}$

We define the following operators on questions that will be useful in the sequel:
a) **Body**(Q) that returns the body of Q;
b) **Qualifiers**(Q) that returns the list of categorial qualifiers;
c) **Variables**(Q) that returns the set of question variables.
As it happens with declarative formulas, also questions can be transformed using substitutions:

$$Q' = Q\theta.$$

Definition
Let θ be a substitution and Q a question.
Qθ is the question such that:
Variables(Qθ) = (**Variables**(Q) - the set of variables substituted by constants) θ
Qualifiers(Qθ) = **Qualifiers**(Q)θ
Body(Qθ) = **Body**(Q) θ
If **Variables**(Qθ) = ∅ then Qθ is a first kind question (i.e. the symbol specifying the kind is not present).

Even thought **Qualifiers**(Q)θ is an abused notation, we use it here for sake of clarity. It should be also noticed that closed substitutions produce yes/no questions. For example the query "On which circumstance do Vance and Begin meet?", represented by the question

? s[circumstance(s)]S{pack(s,meet(Begin,Vance))}

can be transformed using the substitution

θ=**{s** ← DiplomaticVisit**}**

in the question

Qθ=
DiplomaticVisit[circumstance(DiplomaticVisit)]
S{pack(DiplomaticVisit,meet(Begin,Vance))}

i.e. "Does a diplomatic visit in which Vance and Begin meet exist?"

The concept of answerhood we consider is like the one in the logic programming.

Definition
Let Q be a question and θ a substitution. θ generate an answer to Q iff **Body**(Q)θ is deductible from KB with all qualifiers in **Qualifiers**(Q) θ. In the latter case **Qualifiers**(Q) θ ∧ **Body**(Q)θ is the answer to Q generated by θ.

Notice that **Qualifiers**(Q) θ has to be interpreted as a conjunction.
Because of the correspondence between substitutions and answers, when no confusion occurs, we can use indifferently the two terms meaning answer.

Example:
The question
?x [Food(x)]S(occurs(eat(Mary,x), today)
could be answered by the substitution
θ = {x←pizza }.
The corresponding answer will be:
Food(pizza)∧occurs(eat(Mary, pizza),today).

In our approach, a computational one in his nature, soundness of a question is tied to the existence of an answer in the KB. Therefore, a question can be sound with respect to a KB and not sound with respect to another.

3.2 Syntactical Processing of Questions

It is possible to define when a sentence is more general than another one, as in Lu J. et al (1998):

Definition
Let a' and a'' be two declarative sentences.
a' ≤ a'' if and only if a substitution θ exists such that a'' θ= a'

We consider a relation between questions, we call it transformation and denote it by Q ⊢Q' that is like erotetic implication, but is performed syntactically.
From now on, Ans(Q) will be the set of possible answers to question Q.

Definition
Let Q' and Q'' be two questions. Q' is less general than Q'' (denoted by Q $\vdash$Q') iff
$\forall a'' \in Ans(Q'') \exists X \subset Ans(Q') (\forall a' \in X (a' \leq a''))$
The answers to Q'' are called partial answer to Q'.

Obviously, it is not necessarily true that
$\forall a' \in Ans(Q') \exists a'' \in Ans(Q'') (a' \leq a'')$

Example
The answers to the query "On which circumstance do Vance and Begin's wives meet?" are more specific than the answers to the query "On which circumstance do diplomats' wives meet?"

Now we show how the relation $\vdash$ between questions can be exploited to solve two kinds of problems: the *construction of plausible answers* and the *multiple answers selection*.

The first problem arises when a search in a KB gives no answer, but it is possible to build a plausible answer. Firstly, we observe that from the previous definition it follows that if Q' and Q'' are two questions such that Q' $\vdash$Q'' and A is an answer to Q'', also Q'A is a question. This kind of question can be useful in the search process. Indeed, answering an implicated question restricts, sometimes drastically, the search, as the following result shows.

Lemma
Let Q be a question and A a partial answer to Q. If an answer to the question QA exists, then an answer to Q exists too.
Proof: we observe that if θ is an answer to QA, by the properties of substitutions, Aθ is an answer to Q and the lemma holds.

This lemma enable us to build plausible answers applying repeatedly the following steps:

- to compute an implied question;
- to retrieve the answer to the implied question;
- to apply this answer to the original question in order to restrict the search.

We consider for example the well known query about the existence of a meeting between Vance and Begin's wives taken from (Schank, 1986) or (Kolodner 1983a, b) in order to show how the transformation process can be obtained in a syntactical way. In the example (slightly simplified to make it more readable), questions are indicated by a Q, declarative sentences by a D and answers by an A or PA (partial answers), capital letters followed by a number.

Q1)Has Vance's wife ever met Begin's wife?
Q2)On which circumstance do Vance and Begin's wives meet?
Q3)On which circumstance do diplomats' wives meet?
D1) Every time they go with their husbands and they meet, diplomats' wives meet each other.
Q4)On which circumstance do diplomats usually meet?
A4) During international meetings.
PA3) When they go with their husbands to those meetings.
PA2) When they go with Vance and Begin to those meetings.
PA1) Yes, when they go with Vance and Begin to those meetings.
Q5) Do meetings in which Vance and Begin meet and their wives go with them exist?
A5) [NO]

Q6) Which conditions have to hold for Vance and Begin to meet and their wives to go with them?
Q7) Which conditions have to hold for two diplomats to meet?
Q8) Which conditions have to hold for two people to meet?
A8) People need to be in the same place in order to meet.
PA7) Diplomats need to be in the same place in order to meet.
PA6) Vance and Begin need to be in the same place in order to meet.
Q9) Are Vance and Begin in the same place?
A10) [NO. Vance is in U.S.A. and Begin in Israel.]

Q10) Which conditions have to hold for Vance and Begin to be in the same place?
Q11) Which conditions have to hold for two diplomats to be in the same place?
A11) A diplomat usually goes to another country to meet another diplomat.
PA10) Vance needs to go to Begin's country to meet him.
Q12) Has Vance ever gone to Israel?
A12) [List of visit]
(Notice that some further steps would have been necessary to obtain this answer).

Q13) In which of those visits did their wives go with them?
A13) [Particular visits]

It is possible to describe the previous sequences of questions and answers by the following schema, in which $\rightarrow$ relates a question to an answer

Q1$\vdash$Q2$\vdash$Q3D1$\vdash$Q4$\rightarrow$A4$\rightarrow$PA3$\rightarrow$PA2$\rightarrow$PA1$\vdash$Q5 $\rightarrow$[NO]

Q5⊢Q6⊢Q7⊢Q8→A8→PA7→PA6⊢Q9→［NO］
Q5 suspended

Q9⊢Q10⊢Q11⊢A11→PA10⊢Q12→［List of visits]

Q5 reconsidered
Q5 ［List of visits］ ⊢Q13→ ［Particular visits]

This example shows how it is possible to build in a syntactical way a sort of "deductive chain" involving questions whose aim is to search in a KB an answer to a given question. These chains are similar to those called erotetic derivations in (Wisniewski 2003).

The multiple answers selection takes place when a lot of answers are available in the KB and the need of selection arises.

Consider the following query as an example in (Ginzburg 1995a, b):

"Who works in the Philosophy Department?"

Two kind of answers are possible:

1) a general one;
2) a more specific one.

For example

1)A group of neo-positivist philosophers and some erotetic logicians.
2) John X , Mark A, etc.

The first answer is related to a question like

? P [P(x)]S{work(x, Philosophy Department)}

In this case the questioner is not interested in the objects, but in the properties they have. The second kind of answer is related to a question such

? x S(work(x, Philosophy Department))

There can be people interested in the first kind of answer and not interested in the list of names.

Ginzburg (1995a, b) showed that the questioner is not always looking for a specific answer. Indeed, he stressed that, to solve this case, it is necessary to know the goal of the questioner. We believe that the capability of replying to a question by another question could be useful.

We propose the following solution:

-every query must be translated in a more general question as possible ;

-the output is the answer to this question and some queries asking the questioner for further specifications.

An example of reply to the previous question could be: a group of neo-positivist philosophers and some erotetic logicians and the query "Would you like to know the kind of people or just the people's names working in the Department?"

Finally we indicate some way to obtain a more general (or a more specific) question from a given question:

T_1) "*introduction of variable"*

Let Q be a question and c a constant such that P(a) for some predicate P. A more general question Q' is obtained in the following way:

Variables(Q') = **Variables**(Q) U {x};
(x is a new variable not belonging to **Variables**(Q));
Qualifiers(Q') = (**Qualifiers**(Q)θ;
Body(Q') = **Body**(Q) θ.
(θ substitute c with x).

Obviously, the effect of this transformation is a widening of the search field.

T_2) *"search for conditions"*

Variables(Q') = [C];
(a new variable is introduced)
Qualifiers(Q') =**Qualifiers**(Q)U (condition(C));
Body(Q') = (C ⇒**Body**(q)).

4 THE JUMP OF DOMAIN

In this section we will deal with a kind of inference which is very common in the ordinary reasoning: it is something beyond the modus ponens and the syllogism, which considers at least a chain of two implications like ($x \to y$, $y \to z$) ¶ $x \to z$. Another scheme of reasoning is the so called "trial based reasoning". The common support is the concepts-predicate table, which crosses asserts and predicates. We can drag predicates by chaining concepts, provided there is a medium concept which joins two other concepts.

A simple dissimilarity measure between two concepts may be the inverse of the owned predicates, so that if two concepts do not share any predicate it means that between them there is an hole. The material implication between two asserts can be interpreted as the transfer of a predicate ownership from the explanans to the explanandum. Sharing a common predicate means that both the asserts can be chained by an ***if…then***.

It is also easy to see that the implication is merged into an hyperbolic metric structure so that $d(x,y) - d(x,z) > d(z,y)$. This means that the join of two concepts which share only one predicate admits the jump to a third concept. The triangular relation does not hold. As a consequence, it is built in a priority between two predicates. The identification of a cause after a trial implies that the simultaneous occurrence of a subset of predicates is definitively higher than elsewhere in the table of observed cases. From the metric point of view, the product of an inverse function is normalized by the number of occurrences of each predicate among the set of considered concepts. The chain of the product of the inverse of the number of occurrences of n predicates and m concepts means that we can go along a backward path which can simulate the so called explanation of an evidence. In the ***and*** relation no priority is supposed. The hole cuts down any direct relation, represented by an edge between two predicates. We can overcome the stop by a sequence of transfers, so that we can built up an assert which joins two distant predicates and therefore two distant concepts.

For example, every man can be affected by a disease and is mortal. A fail in a machine is like a disease.

We can make a bridge so that we can built up a chain: every machine is mortal.

We can built examples whose chain can be pretty long just using *and* or *if…then*:

i. *young(P_1)people(Π) are itching(P_2) to fight(P_3);*

ii. *young people like(P_0)driving(P_4)cars(P_5).*

We can join *itching to fight* with *like driving cars*. We can write :

iii. *people itching to fight like driving cars.*

This seems to be a common way of thinking.

We can also consider the reverse path, so that we can assert:

People who like driving cars are itching to fight.

The reverse reasoning is allowed here as far as we do not have forbidden directions in any edge of the chain. Through young(P_1)people(Π) we drag itching(P_2) to fight(P_3) and join itching (P_2) to fight(P_3) to like (P0) to driving cars (P5).

The join brings to a new concept: the young drivers are aggressive people.

We propose here the algorithm **BTKSA** based on the concept-predicate table.

The algorithm **BTKSA** is based on the *Ariadne's thread* logic in the search of the exit in the labyrinth.

Input: a question in the form

$?x_1..x_m[P_1(\underline{x})]\ldots[P_m(\underline{x})]\Sigma\{A(\underline{x})\}$

/*It is supposed that the leading predicate exists*/

Step 1. a partial **DB** is rescued by the leading predicate **A** in the query, which becomes the root of a tree T;

Step 2. all the subtrees whose roots are on the tree generated by **A** are identified in the partial **DB**;

/*the edges connecting vertices already connected in the graph are deleted in order to avoid cycles.*/

Step 3. the asserts joining n-tuple of the predicates are identified for n = 2,3,…,k, where k is max length of the branches of the subtrees;

Step 4. all the asserts of the local **DB** are reduced to branches of at least a subtree of the main tree;

Step 5. a scanning process of the whole tree is performed to identify the branch whose vertices are the same predicates distributed in the same order apart for the quantifier and variables which are substituted by a predicate and a free variable;

Step 6. a jump ¶ is a path in the tree connecting different predicates so that the backward course follows;

/* we recall that a material implication enjoys the transitive property so that:

$p(A)\rightarrow q(B)$, $q(B) \rightarrow w(C)$,¶ $p(A)\rightarrow w(C)$.

A backward path recognizes p(A) as the sources of the jump. In the tree a path joining separately the tree predicates attributes the extension of the asserts containing C assigning them A. */

/* the equivalent statistical reasoning is the so called Bayes theorem, which supports a backward reasoning.*/

/* the negation of the last consequence implies the negation of the whole implication. The sequence of backward reasoning implies the contraction of the chain of implications and the negation of the implication of the negated consequence is also true.*/

Step 7.we built up an assert which contains a truth not belonging to the **DB** as far as it is an induction.

Step 8. stop.

Let consider the following formula:

$$(((B\wedge C)\Rightarrow A)\text{and}(C \wedge E)))\text{¶ } E \qquad (1)$$

The **¶** means a jump as far as we need to connect a formula for jumping to another assert. The jump means only a non evident join which can be interpreted as an implication if we give a direction to the path. Actually there are at most n! paths which are reduced by the incidence matrix. An example of compressed incidence matrix of a concept-predicate table is given in the sequel. Each row can be an assert. The direction can be given by further restrictions, considering at first the concept with <u>min sharing</u>, then deleting it and recycling. There is no guarantee of a total ordering if there are no specific requests given by specific implications between couples of concepts. The predicates of C are transferred to *E*. *E* joins two predicates which are not joined at the beginning of the reasoning. The transfer is assured by the jump of a hole. The Ariadne strategy for moving out of a labyrinth is followed here as far as we avoid to cross again the same square in the table. Moreover, we can use as a strategy both a forward movement and a backward one: from the target of the jump or from the first concept. The choice depends from the target domain as before mentioned.

The connection to *E* is not obvious if we do not join P_3 and P_4 as well as P_2 and P_3.

The strategy on the table is to move both horizontally and vertically. As an example we suppose the following incidence structure connecting concepts to predicates:

(A) contains P_2, P_3, P_4

(D,C,B), share P_1, P_2

(A,C) share P_2
(D,E) share P_4
(C) contains P_1, P_3.
The **jump** allows the following scheme:
from the chain $(E \rightarrow A \rightarrow B \rightarrow C) \wedge C$ ¶ E.
If C is the evidence E therefore drags P_1 and P_2.We see that P_1, P_2, P_3, P_4 are 4 predicates and *A,B,C,D* and *E* are 5 asserts bearing 5 concepts, so that the referred subsets overlap each other and a chain pulls two predicates to E. We see that the path starting from C stops at E if two predicates partially overlap.

The jump connects the asserts sharing at least one of the two predicates. In other terms, the logical connection joins the asserts, while the jump joins the predicates and indirectly the asserts. The jump generates something like an implication: P_1 is a common property for A and B, as well as P_8 and P_9 are common properties for B and C. Finally, P_1 is a common property for C and D. E actually extends to C the properties of D.

The backward reasoning can be ascertained from the concept-predicate table if we set up the network of the predicates between all the couples of concepts. As a consequence, we set up a matrix of accessibility between concepts. The network of relations can be considered for the backward reasoning. The Ariadne algorithm starts from the end of the labyrinth; that is the evidence to get the entrance, which is the remote cause of a chain of causes and concurrent causes. The problem is to avoid to meet the same cause many times. In order to get the root of the tree, starting from the top, it is necessary to cut an edge whose accessibility measure is identified by the measure at the right of the inequality. If r and s are number of predicates shared by two overlapping concepts, x and y, respectively, we have the following relation between their segregation evaluated in terms of the inverse of the number of owned predicates:

$$\frac{1}{r} - \frac{1}{s} > \frac{1}{r-s} \qquad (2)$$

$$\frac{1}{r} - \frac{1}{r-s} > \frac{1}{s} \qquad (3)$$

where $r < s$.
The formula looks like a probabilistic formula:

$$P(x) - P(x \cap (\neg y)) > P(y) \qquad (4)$$

where x and y are the concepts whose extentions are r and s, respectively and P is the inverse of the extentions r and s.

There is some difference in the underlying logic background. In a concept-predicate table we have:

$$x \wedge \neg (x \wedge \neg y) \rightarrow y \qquad (5)$$

in other terms:

$(x \vee y) \rightarrow y$.

In probability we usually have:

$$P(x) + P(y) - P(x \cap y) > 0 \qquad (6)$$

In probability x and y are events while here may be the extension of concepts as well as of conceps in terms of number of predicates.
The sound difference here is that formula (2) can be rewritten as

$$\frac{1}{r} + \frac{1}{s} > \frac{1}{s-r} \qquad (7)$$

The geometry is quite different just as the underlying logic. The consequence is that the chaining supposes $y \subset x$, so that the concept x must include y. In other terms, the predicates of y are the predicates of x but it is not true the viceversa. The backward chaining is therefore allowed provided that the previous condition is satisfied. Cutting a circle means cutting the superposition between two concepts. The path from the evidence to the cause is therefore ruled by the previous inequalities.

5 CONCLUSIONS AND FURTHER WORKS

In this paper we first redefine some concepts of erotetic logic like question implication and then we show how they can be used to search information in a KB. The result is the base for a more powerful search engines. Future works will involve an implementation of the system.

REFERENCES

Bellacicco A., 2002. A New Tool for Textual Database Analysis and Management, in Brebbia C.A., Pascolo P. (eds.) *Management Information Systems 2002: GIS and Remote Sensing.* Wit Press.

Bellacicco A., 2003. Two automata linguistic communication system (TALCS) in *7th world* *multiconference on Systemics, Cybernetics and Informatics.* Orlando, Florida, USA.

Bell J.,1995. Changing attitudes. In Wooldridge M. and Jennings N.R. (eds.) *Intelligent Agents, Post-Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages.*

Ben-Ari M., 1993. *Mathematical Logic for Computer Science.* Prentice Hall

Burhans D.T. Expanding the Notion of Answer. *Proceedings of the Grace Hopper Conference, Vancouver*

Cautiero G., Sessa M.I., Sessa M., and Vacca M, 1991. Conceptual Processing of Texts in Paleography. *Information Processing and Management.* 27(2/3), 219-227.

Frauenfelder U.H, 1996. Computational models of spoken word recognition in Dijkstra T. and de SmedtK. (eds.) *Computational Psycholinguistics: AI and connectionist models of human language processing*. Taylor & Francis, Great Britain.

Ginzburg J., 1995a. Resolving questions I. *Linguistics and Philosophy*, 18(5), 459-527.

Ginzburg J., 1995b. Resolving questions II. *Linguistics and Philosophy*, 18(6), 567-609.

Groenendijk J., Stochow M., 1997. Questions. in Van Benthem J., Ter Meulen A. (Eds.) *Handbook of logic and language*. Elsevier, Amsterdam.

Kolodner, J. L., 1983a. Maintaining organization in a dynamic long-term memory. *Cognitive Science*, 7, 243-280.

Kolodner J.L., 1983b. Reconstructive Memory: A Computer Model. *Cognitive Science*. 7, 280-328.

Krifka M., 2001. For a Structured Meaning Account of Questions and Answers. In Fery C., Sternfeld W. (Eds.) *Audiatur Vox Sapientia. A Festschrift for Arnim Von Stechow,* Akademie Verlag, Berlin

Lebowitz M., 1988. The Use of Memory in Text Processing. *Commun. ACM* 31(12), 1483-1502

Loia V., Luongo P., Senatore S, Sessa M.I., 2002. Info-Miner: Bridging Agent Technology with Approximate Information Retrieval. In Nikhil R. Pal, Michio Sugeno (Eds.) *Advances in Soft Computing - International Conference on Fuzzy Systems*. Lecture Notes in Computer Science 2275, Springer, Berlin.

Lu J., Harao M., Hagiya M., 1998. Higher Order Generalization. in Dix J., Fariñas del Cerro L., Furbach U. (Eds.) *Logics in Artificial Intelligence, European Workshop.* Lecture Notes in Computer Science, Springer, Berlin.

Ostrovskaya Y.A, 2003. Recognizing text structures: computational applications of formulaic theory of language. in *7th world multiconference on Systemics, Cybernetics and Informatics*. Orlando, Florida , USA.

Piwek P., 1997. The construction of Answers. In Benz A., Jaeger G. (Eds.) *Proceedings of MunDial: the Munchen Workshop on the Formal Semantics and Pragmatics of Dialogue*, CIS- Bericht 97-106 Department of Computational Linguistics, University of Munich.

Ram A., 1991. A theory of questions and question asking. *The Journal of the Learning Sciences,* 1 (2/3), 273-318.

Schank R.C., 1982. *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press.

Schank R.C.,1986. *Explanation Patterns: Understanding Mechanically and Creatively.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Wisniewski A.,1994. Erotetic Implications. *Journal of Philosophical Logic*, 23, 173-195.

Wisniewski A., 1995. *The posing of questions. Logical Foundations of Erotetic Inferences.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Wisniewski A., 1996. The logic of questions as a theory of erotetic arguments. *Synthese*,109,1-25.

Wisniewski A., 2003. Erotetic search scenarios. *Synthese*, 134, 389-42.