Dynamic Generation of Cooperative Natural Language Responses in WEBCOOP

Farah Benamara
IRIT
Toulouse III University
benamara@irit.fr

Patrick Saint Dizier
IRIT
Toulouse III University
stdizier@irit.fr

Abstract

We present in this paper a formal approach for the dynamic generation of cooperative NL responses in WEBCOOP, a system that provides intelligent responses in French to natural language queries on the Web. The system integrates reasoning procedures and NLG techniques paired with hypertext links. Content determination is organized in two steps: providing explanations that report user misconceptions and then offering flexible solutions that reflect the cooperative *know how* of the system.

1 Introduction

The main aim of WEBCOOP is to explore, develop and evaluate reasoning procedures and language technologies for the development of a system that provides intelligent cooperative responses in French to natural language queries on the Web. Besides the use of language at several levels (processing queries, generating responses, extracting knowledge from web pages), such a system requires the integration of knowledge representation and the use of advanced reasoning procedures. Moreover, the complexity of reasoning procedures must be kept reasonable in order to optimise tractability, efficiency, and re-usability.

In a first stage, the project is developed on a relatively limited domain that includes a number of aspects of tourism (accommodation and transportation, which have very different characteristics on the web). WEBCOOP is a direct question-answering system, it is not settled within a dialogue perspective and does not include any user model.

Our challenge is to develop a formal framework that integrates reasoning procedures with real-life data extracted from web pages in order to produce web style cooperative NL responses that reflect the accuracy of the reasoning procedures. Our ultimate goal is to develop a formal pragmatics for cooperative responses.

In this paper, we focus (1) on formal aspects of content determination and (2) on the dynamic and interactive generation of cooperative NL responses by integrating NLG techniques with hypertexts (Dale et al, 98). NL responses are produced from logical formulas constructed from reasoning processes. During the content determination stage, the system has to decide, via cooperative rules, what is relevant and then to organize it in a way that allows for the realization of a coherent response. Hyperlinks are dynamically created at generation time. This leaves up to the user the high-level planning tasks inherent to NLG (Reiter and Dale, 97) and improves readability and information access. The standard generation difficulties (lexicalisation, aggregation (Reape and Mellish, 99), argumentation) remain crucial to generate cooperative responses, but their web-style greatly reduces the overall complexity.

In the following sections, we first present a general typology of cooperative responses followed by a few motivational examples that explain the main current aspects of WEBCOOP. Then we analyse the content determination process, organized in two steps: production of explanations that report user misconceptions and then production of flexible solutions that reflect the cooperative *know how* of the system in order to provide help to the user. Finally, central elements of the surface generation component are briefly presented.

2 Motivations

(Grice, 75) maxims of conversation (*quality, quantity, relation, style*) are often used as a basis for designing cooperative answering systems. These systems are typically able to provide general, descriptive answers along with explanations about these answers. They can respond intelligently to false presuppositions and to various types of misconceptions. They can also relax constraints in a question and provide summaries or conditional responses. A number of cooperative systems were designed in the past for databases such as *COBASE* (Minock and Chu, 96) and *CARMIN* (Chakravarthy et al, 90). Most of the efforts were concentrated on fundamental reasoning procedures, while very little attention was paid to question analysis and to NL response generation. For an overview see (Gaasterland et ali, 94).

2.1 A General Typology of Cooperative Responses

Gricean maxims describe fundamental properties of cooperative behaviours. To address this, specific cooperative techniques have been developed and different types of cooperative responses have then been identified. We have structured and classified these types within a functional architecture. In WEBCOOP, cooperativity can be summarized as follows :

First, the elaboration of cooperative responses is based on the use of reasoning procedures that construct :

(1) explanations of a query failure when false presuppositions or misconceptions in the question conflict with knowledge in the database (figure 1 section 2.2.1) (Gal, 88),

(2) conditional responses reflecting relaxation procedures (Gaasterland et al, 92) when the system cannot find any response (figure 2 section 2.2.1),

(3) explanations related to the interpretation of fuzzy terms (figure 3 section 2.2.1),

(4) warnings indicating the temporal dependency of the response especially for elements with a high temporal variability, intrinsic or observed, such as seat availability or fares in air ticket reservation,

(5) conditions directly related to conditional knowledge in the knowledge base (KB) (Burhans and Shapiro, 01) such as constraints associated with the possibility of a service,

(6) textual information from web pages, describing procedures, definitions or causes, obtained from the query evaluation on the knowledge base of indexed web pages such as asking for visa formalities to enter a given country.

Next, the organisation of the informational content of the response before NL generation includes:

(7) cleaning redundant elements,

(8) providing intentional responses which are abstract descriptions of large sets of extensional responses (figure 3) (Burhans, 02).

(9) summarizing, sorting explanations or focussing on a particular one, using heuristics that apply to sets of integrity constraints, to produce synthetic responses (Gal, 88).

In this paper we focus on the content determination and the surface generation of (1), (2) and, to some extent, of (3).

2.2 Cooperative Responses in WEBCOOP

In WEBCOOP, user's questions may range from keywords to comprehensive natural language expressions. Responses provided to users are built in web style, by integrating natural language generation (NLG) techniques with hypertext links to produce ”dynamic” responses. Responses are structured in two parts. The first part contains explanation elements in natural language. It is a first level of cooperativity that reports user misconceptions in relation with the domain knowledge. The second part is the most important and the most original. It reflects the *‘know-how’* of the cooperative system, going beyond the cooperative statements given in part one. It is based on several compo-

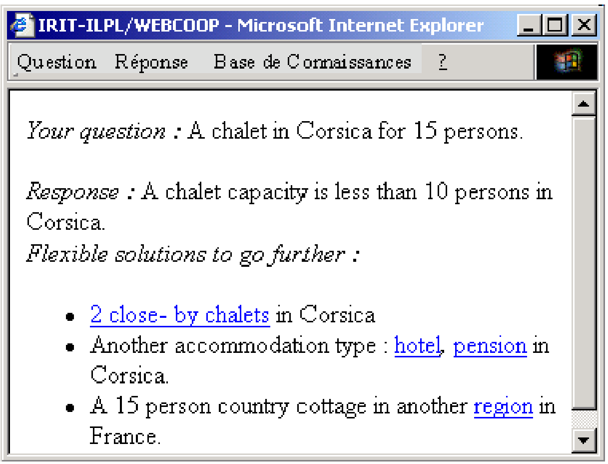

Figure 1: Example 1.

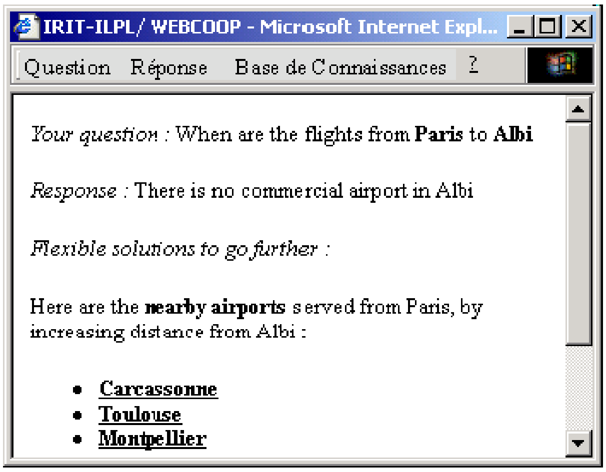

Figure 2: Example 2

nents: dedicated cooperative rules possibly using knowledge extracted from web pages, relaxation strategies and the domain ontology. The *know-how* component also allows for the dynamic determination of those text fragments to be defined as hypertext links, from which the user can get more information. This allows the user to control navigation within the response hyperlink network and, therefore, the general planning of the response.

2.2.1 A Few Examples

To better characterise our problem, we collected a corpus of question answer pairs in French that reflects different cooperative behaviours. The three examples below are extracted from this corpus. For the sake of readability, they are translated into English.

- **Example 1** : Suppose one wishes to rent a 15 person country cottage in Corsica and (1) that observations made on the related web pages or (2) that a constraint or a regulation, indicate that the maximum capacity of a country cottage in Corsica is 10 persons (figure 1).

The first part of the response relates the detection of a false presupposition or the violation of an integrity constraint for respectively cases (1) and (2) above. Case (2) entails the production of the following message, generated by a process that evaluates the question logical formula against the knowledge base: *A chalet capacity is less than 10 persons in Corsica.*

In a second step, the *know-how* of the cooperative system generates a set of flexible solutions as shown in figure 1, since the first part of the response is informative but not really productive for the user. The three flexible solutions proposed emerge from *know-how* cooperative rules based on minimal *relaxation procedures*. The first flexible solution is based on a cardinality relaxation, while in the last two solutions relaxation operates gradually on concepts such as the type of accommodation (hotel or pension) or the region, via the domain model and the ontology. Dynamically created links are underlined. The user can then, at will, get more precise information, dynamically generated from the data base of indexed web pages.

- **Example 2** : In the second example, the user asks for flights from Paris to Albi. The user's false presupposition is detected and a cooperative response is first generated by indicating that *there is no commercial airport in Albi* (figure 2). The *know-how* component goes further by providing responses using first relaxation procedures and then *scalar implicature* to rank the nearests airports served from Paris, by increasing distance from Albi. The airports names are generated as dynamic links in order to get flight information.

- **Example 3** : Another type of cooperativity is a flexible interpretation of fuzzy terms in questions (Figure 3). Suppose the user asks for : *a country cottage not too close to the sea in Côte d'Azur*. Since this question does not contain any false presupposition, the *know how* component produces

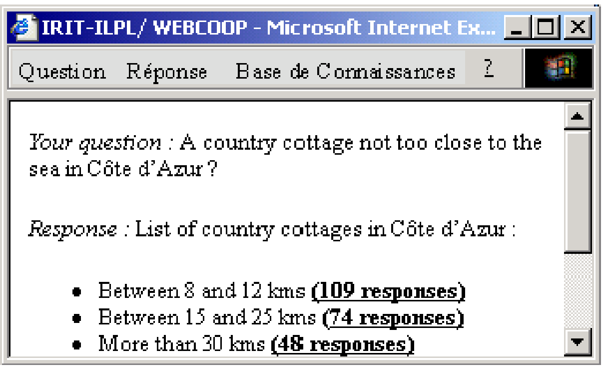

Figure 3: Example 3

a direct response. It is elaborated using, in our case, techniques based on geometrical considerations which compute the distance from each country cottage to the sea and group cottages according to dynamically determined distance intervals. When the user clicks on an interval, since there are many responses, the *intentional component* produces a synthesis of responses according, for example, to the localisation of country cottages such as *mountain*, *country side*, *river bank* which are at the same conceptual level as *sea side*.

3 Reasoning Procedures for Content Determination

3.1 Detection of Misconceptions

Misconceptions include, among others, a variety of false presuppositions. A false presupposition occurs when a user presupposes the existence of an entity that does not exist or presupposes a relation between entities (or entity type) that is inconsistent with the knowledge base. The detection of these phenomena in a question implies the generation of an explicative response that reports the conflict between that question and the knowledge base.

Formally, presuppositions in a logic conjunctive query, written in a *variable substituted form* (Gal, 88), are all formulas derived from the original query by removing one or several literals. Meaningful presuppositions are those which are *conceptually and logically coherent expressions*. The meaningful subsets of presuppositions are structured as a partial order using the generalisation relation as an order relation.

One of the most studied techniques in the literature for the detection of false presuppositions (Gaasterland et al, 96), is the *merge compatibility* between the query and the integrity constraints of the knowledge base. The merging process produces residues which allow for semantic information to be added to the query without changing its denotation. If the residue contains an incoherence, then a false presupposition is detected (table 1).

The Knowledge Base
Rule : $cottage(x) \rightarrow chalet(x)$
Constraint : $Fail \Leftarrow cottage(x) \wedge capacity(x, nb)$ $\wedge nb > 10$.
Interpreted User Question
Chalet in Corsica for 15 persons?
$(Entity, x : chalet, in(place, x, Corsica) \wedge$ $capacity(x, 15))$
Deduction Process
Question variable substituted form :
$chalet(x) \wedge in(place, x, y) \wedge capacity(x, nb) \wedge$ $nb = 15 \wedge y = Corsica$.
Merging the rule and the constraint produces:
$Fail \Leftarrow chalet(x) \wedge capacity(x, nb) \wedge nb > 10$
Merging the question and the new constraint produces :
the incoherence : $nb = 15 \wedge nb \leq 10$
Generated Logical Response
$chalet(x) \wedge capacity(x, nb) \wedge nb \leq 10$
-Table 1-

In table 1, the knowledge base is first augmented with a new integrity constraint by merging a rule with an existing integrity constraint. Then merge compatibility can optimally be used involving the query and that new constraint. After detection of an incoherence, the following natural language response is produced: *a chalet capacity is less than 10 persons in Corsica*.

Another kind of incoherence is detected from the failure of a search in the extensional database. This is illustrated in the question : *A flight from Paris to Albi?* of example 2 where $airport(Albi)$ is not a fact of the knowledge base.

If more than 2 integrity constraints are violated, the system has to manage possible redundant explanations and to organize independent explanations. For that purpose, (Gal, 88) proposes heuristics to produce synthetic responses.

3.2 Content Determination in the Know How Component

The cooperative *know how* component aims at providing flexible solutions to go further (1) when there is a misconception as above, (2) when the set of extensional responses is too large or empty, or (3) simply when the response needs an elaboration to be useful. This component is based on intentional description techniques (not treated here) and on intelligent relaxation procedures going beyond classical generalization methods. This component also includes additional dedicated cooperative rules that make a thorough use of the domain ontology and general knowledge. This component has potentially a large number of capabilities. We focus here on a basic, but frequently used reasoning schema: the proximity relation that pairs accurate relaxation techniques with ontological knowledge. We first present the reasoning aspects (for content determination) and then the main natural language generation aspects.

Very roughly, a relaxation is a process that rewrites a query in some way to extend its denotation. (Gaasterland et al, 92) define three types of relaxation techniques : a rewrite of a predicate, the broadening of the domain of a variable or breaking a join dependency. In our approach, the notion of answer in the neighbourhood of a question requires that the data base must include a representation of the most salient object properties and their possible interactions. We present in this paper a relaxation technique based on a generic **proximity relation** that uses inherent properties of objects, a conceptual ontology and lexical semantics relations. This proximity relation applies to different ontological or technical domains such as : distance (example 2), fares, capacity (example 1), type of transportation... The proximity relation is associated with constraints (e.g. minimal relaxations, conceptually graded relaxations, which are relevant for the intentional component as well) to produce information useful and relevant for the user. It is one of the most productive know-how rule. It is an iterative process running till a flexible solution is found that leads to a non-empty and coherent solution.

Similarly to the merge compatibility, the proximity relation is a rewriting rule that operates on a formula. We now present its main logical features.

In the next paragraphs, we assume that the question is written in its variable substituted form (table 1). Let T be the set of variables at stake in the residue. Let X be the set of variables that appear in the predicates where an element of T occurs, and Var=$T \cup X$. Then, let F(X,T) be the set of predicates in the query where at least one of the variables in Var occurs, and R the remainder of the query. The general rewriting rule is:
$proximity(F(Var), T) \wedge R \rightarrow NewFormula$,
where $NewFormula$ realises several generic operations, among which:

1. considering several objects of the same type, instead of just one, till the constraint is satisfied. F(X,T) is duplicated to introduce several instances of the same object T till the desired quantity of T is reached (example 1),

2. proposing the same kind of resource but with a ranked set of values close to the initial value at the origin of the failure (example 2),

3. relaxing on the resource via the minimal generalisation that makes consistent the previously inconsistent user constraint, e.g. via the least upper bound in the generalizations lattice (example 1).

The notion of proximity is implemented by the predicate $near(CD, V, Y, Result)$ where CD is a type in a conceptual domain, Y is a resource resulting from the relaxation, of the same type than V. V is the resource on which relaxation operates. Result contains the sorted set of results according to the criteria associated with the conceptual category. W.r.t. the three operations described above, *NewFormula* has the following forms:

1. $F(X1, T1) \wedge F(X2, T2) \wedge X1 \neq X2 \wedge near(CD, X1, X2, Result) \wedge R \wedge (T1 + T2 \geq T)$ (T1, T2 and T contain a single variable).

2. $F(X, T) \wedge near(CD, T : Type, Y : Type, Result) \wedge T \neq Y \wedge R$. T and Y are of the same type.

3. $near(typeof(V), V : Type, Y, Result) \wedge F'(X, V) \wedge R$.
   F'(X,V) is F(X,V) without the predicates typing X. V belongs to X.

The instantiation of these formulas can respectively be illustrated by the following examples :

1. For example 1 in figure 1, the rewrite of the question formula is :
   $chalet(x) \wedge in(place, x, z) \wedge capacity(x, nb1) \wedge chalet(y) \wedge in(place, y, z) \wedge capacity(y, nb2) \wedge x \neq y \wedge near(place, x, y, result) \wedge z = Corsica \wedge (nb1 + nb2) \geq 15$,
   which proposes in *result* a set of two nearby cottages x and y instead of one.

2. The query :
   $flight(x, paris, T) \wedge T = albi$,
   in example 2, figure 2, is relaxed into
   $flight(x, paris, T) \wedge T = albi \wedge near(place, T, z, result) \wedge T \neq z$,
   which computes the nearests airports from Albi served from Paris.

3. The logical formula of example 1:
   $chalet(x) \wedge in(place, x, y)$
   $\wedge capacity(x, nb) \wedge nb = 15 \wedge y = corsica$
   is rewritten into :
   $near(typeof(x), x, v, result) \wedge in(place, v, y)$
   $\wedge capacity(v, nb) \wedge nb = 15 \wedge y = corsica$
   according to the domain ontology and the minimal generalization strategy. This allows us to propose objects *v* of a type close to x, e.g.: hotels or pensions (listed in the variable *result*) that can accommodate more than 15 persons in Corsica.

   In the know-how component, a response is often composed of an ordered sequence of proximity rule applications, starting with those responses which are the closests to the initial query. A priori, rules 1 to 3 above are organized by increasing generalizations.

4 Dynamic Surface Generation of Cooperative Responses

Our aim is to maximize over the hyperlinks network (Dale et al, 98) the cognitive as well as the communicative coherence of responses provided to the user. At our level, the main advantages are: (1) to leave up to the user the high-level planning tasks inherent to NLG and (2) to improve readability and information access. Each search may result ultimately in the display of a series of web pages related to the initial query.

4.1 Generating Misconception Reports

Generating misconception reports is relatively simple from a linguistic and NLG point of view when a single false presupposition or integrity constraint violation is detected. The main difficulties are: (1) the generation of expressions in the scope of a negation and (2) lexicalisation. Concerning negation, our strategy in the production of cooperative responses is to keep the scope of negation operators minimal, avoiding thus ambiguities. In that case, either generation is based on the equivalent contraposed form of the negated expression (e.g. not cheaper becomes more expensive) or, when impossible, a negation is generated. In this latter case, it often has the form *there is no, it is not possible to, etc.*. About lexicalisation, our strategy is to keep track of the terms used in the related query and to use them as much as possible. A particular case of lexicalization is the lack of a predicate corresponding to a verb in the formula. Our strategy is then to search for a predicative noun corresponding to a deverbal predicate (or a synonym) with the largest set of arguments possible (to guarantee its central relational role) and to lexicalise it as a verb. For example, cout(x,y) is lexicalized as the verb *coûter* (cost).

Organizing reports with more than one violation is more subtle and requires e.g. redundancy elimination (Gal, 88), ordering with adequate connectors, and pronominalization. Concerning explanation ordering, we are exploring heuristics where the formula which has the largest ratio 'number of free variables / number of predicates' is prefered because of its generality. Interestingly, this technique is the opposite of query optimization in data bases.

Let us now consider a simple example, the NLG of example 1 in figure 1 : $chalet(x) \wedge capacity(x, N) \wedge N \leq 10$. Generation proceeds roughly as follows: aggregation of capacity(x,N) with chalet(x), where capacity(x,N) is the head noun since it is relational, i.e.: *la capacité d'un chalet*. Then comes the VP represented by $N \leq 10$, N is a reference to the subject, while 10 is the object. $\leq$ lexicalises as *est inférieure à*. The whole sentence is then: *la capacité d'un chalet est inférieure à 10.* (A chalet capacity is less than 10).

4.2 Generating Cooperative Know-How Statements

Let us now explain how cooperative know-how statements are generated. Let us concentrate on the proximity rewriting rules given in section 3. An important feature is that the generation of a statement is based on instantiations of the general schemas of this rewriting rule. The NLG process is based on (1) a few predetermined sentence fragments and (2) on the assembling of underspecified fragments which are instantiated from the types of the objects at stake in the formula being treated and from references to lexicalised nodes in the domain ontology. A statement is generated only when the formula produced by the proximity relation rewriting rule has a solution in the knowledge base. Hypertext links range over generalized concepts (usually NPs) or on any term which is a part of the response that corresponds to a non-terminal concept in the ontology. In a response, concepts underlined as hypertext links are a priori in different branches in the ontology. The choice of hyper text links is also guided by the underlying intentions of the response. We give below the three generation schemas and briefly show their application on an example:

1. $F(X1, T1) \wedge F(X2, T2) \wedge X1 \neq X2 \wedge near(CD, X1, X2, Result) \wedge R \wedge (T1 + T2 \geq T)$.
   The duplication (possibly more than 2) of the formula F entails the generation of the determiner *deux* (two), then the abstract form F(X,V) is treated. $Near(-,-,-,-)$ is lexicalised e.g. as *proches* (depending on the ontological type of CD) and then the constraints in R are generated (this is close to their formulation in the query). Since no explicit list of chalets is given, the first NP is underlined as an hypertext link, pointing to a list which can be produced upon request. For example 1, we get: *Deux chalets proches en Corse* (two close-by chalets in Corsica).

2. $F(X, T) \wedge near(CD, T : Type, Y : Type, Result) \wedge T \neq Y \wedge R$.
   F(X, T) is first generated, using the principles shown in section 4.1. For example 2, where F(X, T) = $flight(id, paris, City)$, the following NL fragment is produced: *les vols de Paris* (Flights from Paris), the destination (City) is not lexicalised since it is a variable. Next, the predicate near() is treated, the ontological domain being localization in example 2, a preposition like *vers* is generated, expressing a notion of destination deduced from the thematic role of the argument Y, which is then lexicalised. This lexicalization is an hypertext link if no explicit list is given. Then, *near()* is lexicalised as *proche de* and then we have the proper noun of the original query *Albi* (in T). The sentence ends by an underspecified text fragment realized as: *par distance croissante d'Albi, sont:* (by increasing distance from Albi are:). The full sentence is: *Les vols de Paris vers des aéroports proches de Albi, par distance croissante de Albi sont:* (Flights to airports near Albi, by increasing distance from Albi are:). Then follows the list of city/airport names underlined as hypertext links (instead of the lexicalization of Y which is more general).

3. $near(typeof(V), V : Type, Y, Result) \wedge F'(X, V) \wedge R$.
   where F' keeps track of the constraint. The predicate near is here realized as: *nous vous proposons le/les* followed by the lexicalisation of *Type*, followed by the generation of the constraint(s) in F' as given in the original query. For example 1, we get: *Nous vous proposons les logements touristiques suivants pour 15 personnes:* (we propose you the following tourist accommodations for 15 persons) followed by a list of tourist accommodations Y elaborated in Result, and underlined as hyperlinks. This constitutes a simple intentional response.

5 Conclusion

We presented an approach to the generation of cooperative NL responses in WEBCOOP, a system that provides intelligent responses in French to natural language queries on the Web. We focused in this paper on formal aspects of content determination and on the surface generation of cooperative responses. The content determination process is organized in two steps: production of explanations that report user misconceptions and then production of flexible solutions that reflect the cooperative *know how* of the system in order to provide help to the user. The *know how* component is actually based on intelligent relaxation techniques using a generic **proximity relation**, which goes beyond classical generalization methods. This component also allows for the dynamic determination of the responses to be defined as hypertext links.

The WEBCOOP project is in an early stage of development and its implementation is underway. Reasoning and language generation procedures are implemented in Prolog (with constraint interpreters), while the external aspects are developed in Java.

This project has obviously has several future directions among which we plan to :

- Develop new cooperative know-how strategies and their related logical expressions and implementations (e.g. for fuzzy terms),
- Analyse how NLG argumentation techniques (Horacek, 99) can be used, particularly when a misconception is based on the interaction of several integrity constraints.
- Specify different strategies for generating intentional responses in the know how component, for example when the number of direct responses is very large.
- Study the external display of textual fragments extracted from web pages, in particular for queries requiring *narrative* responses such as procedures or regulations.

A thorough evaluation of the results is crucial in this project at two levels: the quality of the services offered to a user and the re-usability to other domains identifying where are the difficulties, what are the costs, what is domain specific and, finally, what can be shared.

Acknowledgements

We wish to thank Claire Gardent for her relevant suggestions, and the GENI project group.

References

Burhans D. 2002. *A Question Answering Interpretation of Refutation Resolution*. Phd thesis, Buffalo.

Burhans D. T., and Shapiro, S. C. 2001. *Abduction and Question Answering*. Proceedings of the IJCAI-01 Workshop on Abductive Reasoning, Seattle, WA.

Chakravarthy U, Grant J, and Minker J. 1990. *Logic-Based Approach to Semantic Query Optimisation*, ACM Transactions on Database Systems, 15(2):162-207, 1990.

Dale R., Oberlander J., Milosavljevic M., Knott A. 1998. *Integrating Natural Language Generation and Hypertext to Produce Dynamic Documents*, Interacting with Computers 11(2), pp. 109-135.

Gal, A. 1988. *Cooperative Responses in Deductive Databases*, Phd Thesis, University of Maryland, Department of Computer Science.

Gaasterland T, Godfrey P, Minker J. 1994. *An Overview of Cooperative Answering*. Papers in Non-standard Queries and Non-standard Answers, in series Studies in Logic and Computation, Clarendon Press, Oxford.

Gaasterland, T., Godfrey, P., Minker, J. 1992. *Relaxation as a Platform for Cooperative Answering*, Journal of Intelligent Information Systems, volume 1, pp : 293-321.

Grice H. 1990. *Logic and Conversation*, In Cole and Morgan editors, Syntax and Semantic, Academic Press.

Horacek, H. 1999. *Generating Deductive Argumentation in Variable Lengths*, Proceedings of the 7th European Workshop on Natural Language Generation, Toulouse, France.

Minock M, Chu W. 1996. *Explanation for Cooperative Information Systems*, International Symposium on Methodologies for Intelligent Systems, 264-273.

Reiter, R. and Dale, R. 1997. *Building Applied Natural Language Generation Systems*, Journal of Natural Language Engineering, volume 3, number 1, pp: 57–87.

Reape, M, Mellish, C. 1999. *Just What is Aggregation Anyway?*, Proceedings of the 7th European Workshop on Natural Language Generation, May, Toulouse, France.