Cooperative Question Answering in Restricted Domains: the WEBCOOP Experiment

Farah Benamara
Institut de Recherches en Informatique de Toulouse, IRIT
118, route de Narbonne,
31062, Toulouse, France
`benamara@irit.fr`

Abstract

We present an experiment for designing a logic based QA system, WEBCOOP, that integrates knowledge representation and advanced reasoning procedures to generate cooperative responses to natural language queries on the web. The system is first developed for the tourism domain. We then examine how and under what conditions this system can be re-used for other domains.

1 Introduction

The current trend in Question Answering is towards the processing of large volumes of open-domain texts (e.g. documents extracted from the World Wide Web). Open domain QA is a hard task because no restriction is imposed either on the question type or on the user's vocabulary. This is why, most of the efforts (Voorhees, 2003) are focused on answering factoid style questions (and to a little extend, definition questions) using shallow text processing which is roughly based on pattern extraction or information retrieval techniques. However, QA should support the integration of deeper modes of language understanding as well as more elaborated reasoning schemas for more complex QA strategies, in order to provide, for example, better answer ranking, answer justification, responses to unanticipated questions or to resolve situations in which no answer is found in the data sources. Cooperative answering systems are typically designed to deal with such situations, by providing non-misleading, and useful answers to a query. (Grice, 1975) maxims of conversation namely the quality, quantity, relation and style maxims are frequently used as a basis for designing cooperative answering systems. An overview of cooperative answering techniques is given in (Gaasterland et al., 1994).

In COGEX (Moldovan et al., 2003), a recent QA system, authors used automated reasoning for QA and showed that it is feasible, effective and scalable. This logical prover aims at checking and extracting all kinds of lexical relationships between the question and its candidate answers using world knowledge axioms, supplied by WordNet glosses, as well as rewriting rules representing equivalent classes of linguistic patterns. Such inference techniques (e.g. lexical equivalence, unification on logical representations of texts) are not sufficient for providing intelligent or cooperative responses. Indeed, advanced strategies for QA requires, as we explain in this paper, the integration of reasoning components operating over a variety of knowledge bases, encoding common sense knowledge as well as knowledge specific to a variety of domains.

We relate in this paper, an experiment for designing a logic based QA system, WEBCOOP, that integrates knowledge representation and advanced reasoning procedures to generate cooperative responses to natural language (NL) queries on the web. This experiment is first carried out on a relatively restricted domain that includes a number of aspects of tourism (accommodation and transportation, which have very different characteristics on the web). The tourism domain is in fact half way between an open domain and a closed domain (e.g. weather forecast, Unix technical manuals). The tourism domain has a kernel roughly around accommodation and transportation, but it also includes satellite domains, such as history, security, health, immigration, ecology, etc. Those satellite domains are only partly considered, from the point of view of the 'kernel' domains. We also observe that there is, in fact, a kind of continuum between the notions of open domain and closed domain, via restricted domains which makes quite fuzzy the definition of what a restricted domain is.

Besides the technical functionalities of WEBCOOP, the main goal of this paper is to evaluate the different facets of the portability of WEBCOOP. Three major points are at stake: (1) resources, in term of language resources and kinds of knowledge required, (2) cooperative procedures involved, such as identifying and explaining user false presuppositions, relaxing constraints or providing intensional responses, and finally (3) the intelligibility of the system outputs (such as hyperlinks, short responses or list of answers), considering that answers should also include a trace of the inferences drawn.

In the next sections, we briefly present the WEBCOOP architecture focusing on the kinds of knowledge and cooperative procedures involved. Then, we analyze the main characteristics of the tourism domain and outline its main features as a restricted domain. Then, we analyze the portability of this type of QA system to other restricted domains. Finally, we propose an evaluation methodology based on experimental psychology for the point (3) cited in the last paragraph.

2 The WEBCOOP Architecture

2.1 A Corpus Based Approach

To have a more accurate perception of how cooperativity is realized in man-man communication, we collected a corpus of question answer pairs (QA pairs) found in a number of web sites dedicated to different kinds of large public domains. 60% of the corpus is dedicated to tourism (our implementation being based on this application domain), 22% to health and the other QA pairs are dedicated to sport, shopping and education. The analysis of this corpus aims at identifying the external form and the conceptual categories of questions, as well as categorizing the different cooperative functions deployed by humans in their discourse. Our main claim is that an automatic cooperative QA system could be induced from natural productions without loosing too much of the cooperative contents produced by humans. We noted that human responses are much more diverse than any machine could produce in the near future. Nevertheless, it is possible to normalize these forms to more stereotyped utterances.

2.2 The Architecture

The general architecture of the system (figure 1) is inspired from our corpus analysis. Our system being a direct QA system, it does not have any user model.

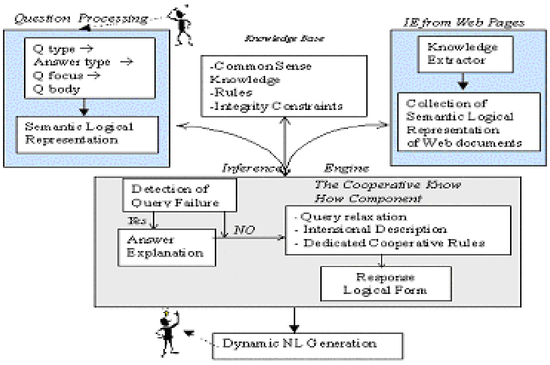


Figure 1: The WEBCOOP architecture

In WEBCOOP, NL responses are produced from first order logical formulas constructed from reasoning processes carried out by an inference engine. Our approach requires the development of a knowledge extractor from web pages (Benamara and Saint Dizier, 2004b) (viewed as a passage retrieval component) and the elaboration of a robust question parser. We assume that the most relevant documents to the user's question are found using standard information retrieval techniques and that the relevant paragraphs that respond to the question keywords are correctly extracted from those documents (Harabagiu and Maiorano, 1999). Then, our knowledge extractor transforms each relevant paragraphs into a logical representation. The WEBCOOP inference engine has to decide, via cooperative rules, what is relevant and how to organize it in a way that allows for the realization of a coherent and informative response.

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 (answer explanation). 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 intensional description techniques and on intelligent relaxation procedures going beyond classical generalization methods used in AI. This component also includes additional dedicated cooperative rules that make a thorough use of the domain ontology and of general knowledge. In WEBCOOP, responses provided to users are built in web style by integrating natural language generation (NLG) techniques with hypertexts in order to produce dynamic responses (Dale et al., 1998).

We claim that responses in natural language must make explicit in some way, via explanations and justifications, the mechanisms that led to the answer. For each type of inference used in WEBCOOP, we define general and underspecified natural language templates (Reiter, 1995) that translate the reasoning mechanisms in accessible terms. A template is composed of three parts, S, F, and R, where :

-S are specified elements,

-F are functions that choose for each concept in the ontology, its appropriate lexicalization,

- R are logical formulas representing the rest of the response to be generated.

The underspecified elements, F and R, depend on the question, on local semantic factors and on the type of solution elaborated. Their generation relies on ontological knowledge, general linguistic knowledge and lexicalisation and aggregation functions. Templates have been induced from a number of QA pairs found in large public domaines. Responses have been normalized without loosing too much of their accuracy in order to get stereotyped response forms usable in NL generation frameworks. A large portion of underspecified elements, within a template, is presented as an hyperlink to the user as illustrated in the examples in the next section. Here is an example of a template dedicated to one of our relaxation schemas. It is used when the question focus is relaxed using its sister nodes in the ontology. Specified elements are in italic:

un autre type de lexicalisation(mother_node): (lexicalisation(sister_node))$^+$ R. [1]

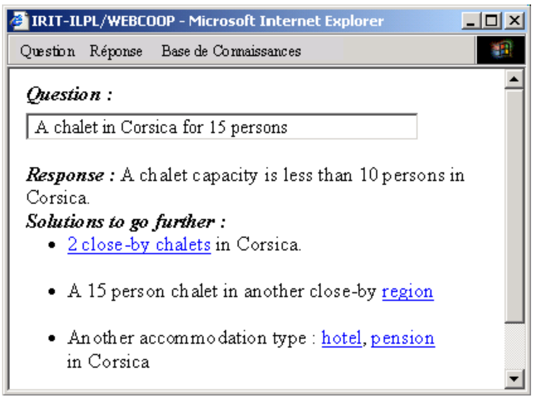
At the moment, in WEBCOOP we have 28 basic templates.

2.3 Two Typical Examples

The following examples illustrate WEBCOOP outputs.

Example1. 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, indicates 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 component of the cooperative system generates a set of flexible solutions as shown in the figure above, 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 relaxation procedures designed to be minimal and conceptually relevant. 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 (possibly a close-by region, with similar characteristics), 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. For technical details on how relaxed responses are elaborated and generated in NL see (Benamara and Saint Dizier, 2004a).



Question :

A chalet in Corsica for 15 persons

Response : A chalet capacity is less than 10 persons in Corsica.

Solutions to go further :

- 2 close-by chalets in Corsica.
- A 15 person chalet in another close-by region
- Another accommodation type : hotel, pension in Corsica

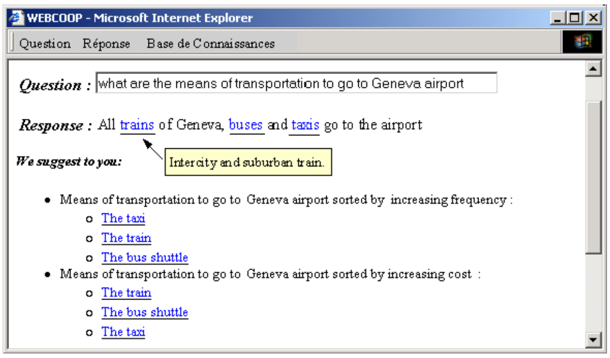
Figure 2: Detection of a misconception and query relaxation

Example 2. Suppose a user asks for *means of transportation to go to Geneva airport.* In WEBCOOP, we have a *variable-depth intensional calculus* which allows us, experimentally, to tune the degree of intensionality of responses in terms of the abstraction level in the ontology of the generalizes. This choice is based on a conceptual metrics that determines the ontological proximity between two concepts. The goal is to have a level of abstraction adequate for the user.

---

[1] A template fragment of the form (fragment)$^+$, indicates that that fragment occurs in the generated response at least one time.

A supervisor manages both the abstraction level and the display of the elaborated intensional answers (IA). The retrieved IA are structured in two parts. First, the generation of a response with generalizations and exceptions: *all trains, buses and taxis go to the airport*. Then, a sorted list of the retrieved extensional answers is generated according to the frequency and to the cost of transportation. This strategy avoids the problem of having to guess the user's intent. For technical details on how IA are elaborated and generated in NL see (Benamara, 2004).



Question : what are the means of transportation to go to Geneva airport

Response : All trains of Geneva, buses and taxis go to the airport

Intercity and suburban train.

We suggest to you:

- Means of transportation to go to Geneva airport sorted by increasing frequency :
    - The taxi
    - The train
    - The bus shuttle
- Means of transportation to go to Geneva airport sorted by increasing cost :
    - The train
    - The bus shuttle
    - The taxi

Figure 3: Variable depth intensional answers

3 Sources of Knowledge and Inference mechanisms in WEBCOOP

3.1 Knowledge Representation for the Tourism Domain: a Typology

A first question about knowledge, for automating the production of cooperative responses, concerns the type and the typology of knowledge involved and where such knowledge can be best represented: in databases, in knowledge bases, in texts (involving knowledge extraction or fragments of text extractions). So far, the different forms of knowledge we have identified are, roughly:

1. general-purpose, factual information (places, distances, proper names, etc.),
2. descriptive information like flight schedules, hotel fares, etc. that we find in general in databases,
3. common sense knowledge and constraints such as: *for a given trip, the arrival time is greater that the departure time*,
4. hierarchical knowledge: such as a *hotel is a kind of tourist accommodation*. This knowledge is often associated with properties that define the object, for example a *restaurant* is characterized by its *type of food, category, localization, etc.*
5. procedures or instructions that describe how to prepare a trip or how to book a room in a given hotel category.
6. definitions,
7. regulations, warnings,
8. classification criteria of objects according to specific properties such as sorting hotels according to their category.
9. interpretation functions, for example, of fuzzy terms (e.g. expensive, far from the beach).

Items 8 and 9 have a quite different nature, but they are closely related to the domain at stake.

3.2 Knowledge Representation in WEBCOOP

Let us now consider how these forms of knowledge are represented. WEBCOOP has two main forms for encoding knowledge: (1) general knowledge and domain knowledge represented by means of a deductive knowledge base, that includes facts, rules and integrity constraints and (2) a large set of indexed texts, where indexes are logical formulae. Our semantic representation is based on a simplified version of the Lexical Conceptual Structure (LCS). Let us review these below.

The kernel-satellite structure of the tourism domain requires that we study, for this application, portability and data integration aspects for each satellite domain. At this level of complexity there is no ready-made method that we can use; furthermore, most of the work is done manually. The results of the integration reflect our own intuitions coupled with and applied on generic data available on the web.

a. The knowledge base is coded in Prolog. It includes basic knowledge, e.g. country names coded as facts or distance graphs between towns, coded as facts and rules. It also includes rules which play at least two roles: data abstraction (e.g. to describe the structure of an object, besides e.g. part-of descriptions found in the ontology):

```
hotel_stay_cost(Hotel_ID, NbNights, Total)
:- hotel(Hotel_ID, Night_rate),
Total is NbNights * Night_rate.
```

and the encoding of conditional situations:

```
book_flight(A) :-
person(A), age(A, AG), AG > 17.
```

which says that you can book a flight if you are at least 18 years old. Finally the knowledge base contains integrity constraints. For example, the constraint:

```
constraint([chalet(X), capacity(X,C), C>
10], fail).
```

indicates that ‘a chalet cannot accommodate more than 10 persons’.

The ontology, described below, contains data which can be interpreted as facts (e.g. hierarchical relations), rules or integrity constraints (as simple as domain constraints for property values). Currently, our KB contains 170 rules and 47 integrity constraints, which seems to cover a large number of situations.

b. The ontology is basically conceptual where nodes are associated with concept lexicalizations and essential properties. Each node is represented by the predicate :

```
onto-node(concept, lex, properties)
```

where *concept* is described using *properties* and *lex* are possible lexicalisations of *concept*. Most lexicalisations are entries in the lexicon (except for paraphrases), where morphological and grammatical aspects are described. For example, for *hotel*, we have:

```
onto-node(hotel, [[htel], [htel,
rsidence]], [night-rate, nb-of-rooms]).
```

There are several well-designed public domain ontologies on the net. Our ontology is inspired from two existing French ontologies, that we had to customize: TourinFrance [2] and the bilingual (French and English) thesaurus of tourism and leisure activities [3] which includes 2800 French terms. We manually integrated these ontologies in WEBCOOP (Doan et al., 2002) by removing concepts that are either too specific (i.e. too low level), like some basic aspects of ecology or rarely considered, as e.g. the economy of tourism. We also removed quite surprising classifications like sanatorium under tourist accommodation. We finally reorganized some concept hierarchies, so that they ‘look’ more intuitive for a large public. Finally, we found that some hierarchies are a little bit odd, for example, we found at the same level *accommodation capacity and holiday accommodation* whereas, in our case, we consider that *capacity* is a property of the concept *tourist accommodation*. We have, at the moment, an organization of 1000 concepts in our tourism ontology which describe accommodation and transportation and a few other satellite elements (geography, health, immigration).

c. The lexicon contains nouns, verbs and adjectives related to the tourism domain, extracted from both corpora and ontologies. The lexicon contains also determiners, connectors and prepositions. The lexicon is constructed directly from the revised ontologies for nouns. Nouns contain basic information (e.g. predicative or not, count/mass, deverbal) coded by hand, their ’semantic’ type, directly characterized by their ancestor in the ontology, and a simple semantic representation. Verbs are those found in our corpora. We have a large verb KB (VOLEM project)(Fernandez et al., 2002) of 1700 verbs in French, Spanish and Catalan. The verb lexicon is extracted from this KB almost without modification. For tourism, including request verbs, we have 150 verbs. Since verbs are central in NLG, it is crucial that they get much information, in our system: thematic roles, selectional restrictions, syntactic alternations, Wordnet classification, and semantic representation (a conceptual representation, a simplification of the Lexical Conceptual Structure).

d. Indexed texts. Our knowldge extractor, which is based on the domain ontology, transforms each text fragment into the following logical representation : `text(F, http)` where F is a first-order formula that represents knowledge extracted (in general) from a web page, with address http (or explicit text). For example, indexed texts about airport transportations in various countries have the following form:

$$text(route(50) \land to(50, cointrin) \land \\ bymeansof(50, Y) \land tramway(Y) \land airport(cointrin) \\ \land localization(cointrin, in(geneva)),\ www.gva.ch).$$

Indexed paragraphs also describe categories such as: procedures, regulations, warnings or classifications. Texts identified as such are indexed by indicating (1) the category in which they fall, (2) a keyword or a formula that identifies the nature of the procedure, regulation, etc., and (3) the text itself, generally used as such in a response.

e. Query representation and evaluation. Processing a query allows for the identification of: the type of the query (yes/no, Boolean or entity, etc.), the question focus and the construction of its semantic representation

---

[2] www.tourinfrance.net

[3] www.iztzg.hr/indokibiblioteka/THESAUR.PDF

in first-order logic. For example, the question: *what are the means of transportation to go to Geneva airport ?* has the following logical representation: $(entity, meansoftransportation(Y),$ $route(X) \wedge to(X, Z) \wedge$ $bymeansof(X, Y) \wedge meansoftransportation(Y) \wedge$ $airportof(Z, geneva))$

Given a fragment of text, we infer that it is an answer to a question by two different ways: (1) from the deductive knowledge base, in that case, responses are variable instances or (2) from the indexed text base, and in that case, responses are formulae which unify with the query formula. In this latter case, roughly, unification proceeds as follows. Let Q (conjunction of terms q_i) be the question formula and F (conjunction of f_j) be a formula associated with an indexed text. F is a response to Q iff for all q_i there is an f_j such that:
(i) q_i unifies with f_j or
(ii) q_i subsumes, via the concept ontology, f_j (e.g. means-of-transportation(Y) subsumes tramway(Y)), or
(iii) q_i rewrites, via rules of the knowledge base, into a conjunction of f_j, e.g.: $airportof(Z, geneva)$ rewrites into: $airport(Z) \wedge localisation(Z, in(geneva))$.

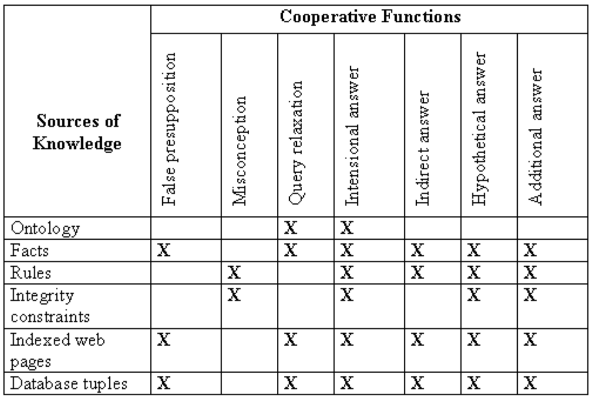
3.3 Inference Needs for Providing Cooperative Responses

We develop a general typology of cooperative functions. The aim is to identify the types and sources of knowledge associated with each of these functions. In terms of portability, we think that annotating in QA corpora of a specific domain the various cooperative functions used should help identify the needs in terms of knowledge for the development of each cooperative function. It remains, then, to evaluate the validity and the adequacy of the inference schemas, but these can only be evaluated a posteriori, whereas the types of knowledge can be evaluated a priori.

Another perspective is that, given the description of the forms of knowledge associated with an application, it may be possible to anticipate what kinds of cooperative functions could be implemented for this application.

We decompose cooperative functions into two main classes: **Response Elaboration (ER)** and **Additional Information (ADR)**. The first class includes response units that propose alternatives to the question whereas the latter contains a variety of complements of information, which are useful but not absolutely necessary such as precision, suggestion or warnings.

Figure 4 shows the different kinds of knowledge involved for each of the cooperative functions that belong to the ER class [4] :



Sources of Knowledge	**Cooperative Functions**						
	False presupposition	Misconception	Query relaxation	Intensional answer	Indirect answer	Hypothetical answer	Additional answer
Ontology			X	X			
Facts	X		X	X	X	X	X
Rules		X		X	X	X	X
Integrity constraints		X		X		X	X
Indexed web pages	X		X	X	X	X	X
Database tuples	X		X	X	X	X	X

Figure 4: Cooperative functions and related knowledge

In the tourism domain, queries are very diverse in form and contents. From that point of view, they are closer to open domains than to closed domains, as advocated in the introduction. Questions about tourism, as revealed by our corpora studies, include false presuppositions (FP), misunderstandings (MIS), concept relaxations (RR), intensional responses (IR). For the moment, we investigate only questions of type boolean and questions about entities and we use the inference schemas: FP, MIS, RR and IR cited above. We think it is important to make explicit in the response the types of knowledge used in the inferences and to show how they are organized and lexicalized. As described in example 1 of section 2.3, the explanation given in italic in the response :*another accommodation type*: hotel, pension, indicates that a relaxation based on the ontological type of the concept chalet was carried out.

4 Evaluation of WEBCOOP

It is clear that an evaluation in the TREC style is not relevant for our approach. We have two forms of evaluations: (1) the evaluation of the portability of the system w.r.t. the forms of knowledge involved and the applicability of the inference schemas and (2) the evaluation of

---

[4] *Indirect responses*, for example: *is your camping close to the highway?*, can be indirectly, but cooperatively responded: *yes, but that highway is quiet at night.*

the linguistic and cognitive adequacy of the responses produced by the system.

4.1 Evaluating System Portability

Porting WEBCOOP to other large-public applications, given the complexity of the system, is quite challenging.

4.1.1 The lexicon and the Ontology

First, we claim that the syntax of questions and the template-based approach used for producing responses are relatively stable. At the language level, the main task is to define an appropriate lexicon, in relation with the domain ontology. This task may be somewhat facilitated by the existence of shared resources, however these are quite rare for French. In general, we observe that some resources are common to all applications (e.g. communication or possession verbs), or prepositions, while others are totally specific, with dedicated senses and usages. Creating an application lexicon is costly, in particular when NL generation is involved. To give an evaluation of the complexity, an application like tourism requires about 150 verbs and about 1800 nouns. Among verbs, 100 are generic verbs, with standard senses. Describing verbs is complex, but their number is quite modest. Most nouns are not predicative, therefore, their lexicon can be partly deduced from the domain ontology.

There are many domain ontologies on the web. Although constructed by domain experts, they turn out not to be necessarily adequate for providing responses to a large public of non-specialists. The main difficulties are to customize these ontologies and to manage their coherence in order to produce a domain ontology which leads to coherent and adequate responses, as explained in section 3.2.

4.1.2 The Inference Schemas

In terms of cooperative functions, our experience is that most applications require the same types of functions, but with various degrees of importance. For example, some application will be subject to more cases of misunderstandings than others, depending, e.g. on the complexity of their associated knowledge and on the type of services expected by users. Similarly, the inference procedures used in WEBCOOP have been designed with a certain level of genericity. They should be portable provided that the knowledge resources of the new domain can be implemented using WEBCOOP format, which is quite generic. But, besides QA annotations, which is a very useful perspective, the adequacy of inferences can only be evaluated a posteriori.

In a future stage, we plan to use what Barr and Klavans (Barr and Klavans, 2001) call component performance evaluation which consists of assessing the performance of system components and determining their impact on the overall system performance.

4.2 Evaluating Response intelligibility

Finally, since WEBCOOP produces responses in NL, some of which on a template basis (different from TREC which simply reproduces text extracts), it is important to evaluate the portability of those templates. We propose a method based on experimental psychology, that aims at evaluating the cooperative responses generated in the know-how component of WEBCOOP. Our methodology involves the following steps:

- Evaluating templates **within a single domain** (tourism in our case). This goal includes two main parts :

1. **intra-templates** which aims at evaluating:

   - **response intelligibility** in terms of (1) the adequacy of the response w.r.t the user intent, and of (2) the justifications and explanations mechanisms provided that led to the answer.

   - the **readability** of the responses in terms of (3) the linguistic surface generation of both the underspecified terms and the different lexicalization choices made within each templates, and in terms of (4) the adequacy of our hyperlinks generation heuristics.

2. **inter-templates** which aims at evaluating:

   - the **display order relevance**. If we go back to the example 1 in section 2.3, the responses are displayed following the inverse reading order of the question constraints i.e. *chalet* is the last concept to be relaxed in the question. This evaluation can also be useful for identifying other kinds of correlation between the answers display and the constraints order in the question.

   - the **general fluency** in terms of syntaxical regularities of the responses generated by each template.

   - the **visual aspect** of the responses : enumerations vs. paragraphs.

- Evaluating **templates portability** to other large public domains like health and education.

We have developed the experimental protocols associated to the relevance of explanation (point 2 cited above) and to the display order relevance. Interpretation results are ongoing.

5 Conclusion and Perspectives

We reported in this paper an experiment for designing a logic based QA system, WEBCOOP, that integrates knowledge representation and advanced reasoning procedures to generate cooperative responses to natural language queries on the web. We claim that restricted domains are more suitable than open domains to conduct research in advanced techniques on QA because those systems require deeper modes of language understanding, more elaborated reasoning schemas paired with a variety of knowledge forms and sources.

WEBCOOP is applied to the tourism domain, it is a challenging and rewarding experience because the tourism domain is half way between open domain applications and closed domains, allowing us to better perceive these two perspectives.

Our corpus based approach allows to identify the type of knowledge associated with each cooperative function. The annotation of corpora constitutes, in our sense, a good evaluation method for the study of the portability of WEBOOP to other restricted domains.

Finally, since an evaluation in TREC style is not relevant for our approach, we have: (1) the evaluation of the portability of the system w.r.t. the forms of knowledge involved and the applicability of inference schemas and (2) the evaluation of the linguistic and cognitive adequacy of the responses produced by the system. We are now evaluating the portability of generation templates to the health and education domains and the accuracy of cooperative functions.

References

V. Barr and J. Klavans. 2001. Verification and validation of language processing systems: Is it evaluation? In *ACL 2001 Workshop on Evaluation Methodologies for Language and Dialogue Systems, July*, pages 34 – 40.

F. Benamara and P. Saint Dizier. 2004a. Advanced relaxation for cooperative question answering. *New Directions in Question Answering, Chapter 21, Mark T. Maybury, editor, AAAI/MIT Press. To appear.*

F. Benamara and P. Saint Dizier. 2004b. Knowledge extraction from the web: an experiment and an analysis of its portability. *Vivek*, 15(1):3–15.

F. Benamara. 2004. Generating intensional answers in intelligent question answering. In *Proceeding of INLG 04, the International Conference on Natural Language Generation, Brighton, UK.*

R. Dale, J. Oberlander, M. Milosavljevic, and A. Knott. 1998. Integrating natural language generation and hypertext to produce dynamic documents. *Interacting with Computers*, 11(2):109–135.

A. Doan, J. Madhavan, P. Domingos, and A. Halevy. 2002. Learning to map between ontologies on the semantic web. In *Proceedings of the 11th international conference on World Wide Web*, pages 662–673. ACM Press.

A. Fernandez, P. Saint-Dizier, G. Vazquez, M. Kamel, and F. Benamara. 2002. The volem project : a framework for the construction of advanced multilingual lexicons. In *Language Technology*. Springer Verlag, Lecture Notes.

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

H. Grice. 1975. *Logic and Conversation*. In Cole and Morgan editors, Academic Press.

S. Harabagiu and S. Maiorano. 1999. Finding Answers in Large Collections of Texts: Paragraph Indexing + Abductive Inference. In *AAAI Fall Symposium on Question Answering Systems, November*, pages 63–71.

D. Moldovan, C. Clark, S. Harabagiu, and S. Maiorano. 2003. Cogex: A logic prover for question answering. In *Language Technology*, pages 87–93. Proceedings of HLT-NAACL, Edmonton.

E. Reiter. 1995. Cogex: A logic prover for question answering. In *NLG versus Templates*, pages 87–93. In Proceedings of 7th European Workshop on Natural Language Generation, Leiden, The Netherlands.

E. M. Voorhees. 2003. Cogex: A logic prover for question answering. In *Overview of the TREC 2002 Question Answering Track*. Proceedings of TREC-11, NIST.