Robust Interaction through Partial Interpretation and Dialogue Management

Arne Jönsson and **Lena Strömbäck***
Department of Computer and Information Science
Linköping University, S - 58183 Linköping, Sweden
email: arj@ida.liu.se lestr@ida.liu.se

Abstract

In this paper we present results on developing robust natural language interfaces by combining shallow and partial interpretation with dialogue management. The key issue is to reduce the effort needed to adapt the knowledge sources for parsing and interpretation to a necessary minimum. In the paper we identify different types of information and present corresponding computational models. The approach utilizes an automatically generated lexicon which is updated with information from a corpus of simulated dialogues. The grammar is developed manually from the same knowledge sources. We also present results from evaluations that support the approach.

1 Introduction

Relying on a traditional deep and complete analysis of the utterances in a natural language interface requires much effort on building grammars and lexicons for each domain. Analyzing a whole utterance also gives problems with robustness, since the grammars need to cope with all possible variations of an utterance. In this paper we present results on developing knowledge-based natural language interfaces for information retrieval applications utilizing shallow and partial interpretation. Similar approaches are proposed in, for instance, the work on flexible parsing (Carbonell and Hayes, 1987) and in speech systems (cf. (Sjölander and Gustafson, 1997; Bennacef et al., 1994)). The interpretation is driven by the information needed by the background system and guided by expectations from a dialogue manager.

The analysis is done by parsing as small parts of the utterance as possible. The information needed by the interpretation module, i.e. grammar and lexicon, is derived from the database of the background system and information from dialogues collected in Wizard of Oz-experiments. We will present what types of information that are needed for the interpretation modules. We will also report on the sizes of the grammars and lexicon and results from applying the approach to information retrieval systems.

2 Dialogue management

Partial interpretation is particularly well-suited for dialogue systems, as we can utilize information from a dialogue manager on what is expected and use this to guide the analysis. Furthermore, dialogue management allows for focus tracking as well as clarification subdialogues to further improve the interaction (Jönsson, 1997).

In information retrieval systems a common user initiative is a request for domain concept information from the database; users specify a database object, or a set of objects, and ask for the value of a property of that object or set of objects. In the dialogue model this can be modeled in two focal parameters: Objects related to database objects and Properties modeling the domain concept information. The Properties parameter models the domain concept in a sub-parameter termed Aspect which can be specified in another sub-parameter termed Value. The specification of these parameters in turn depends on information from the user initiative together with context information and the answer from the database system. The action to be carried out by the interface for task-related questions depends on the specification of values passed to the Objects and Properties parameters (Jönsson, 1997).

We can also distinguish two types of information sources utilized by the dialogue manager; the database with task information, T, or system-related information about the application, S.

* Authors are in alphabetical order

3 Types of information

We can identify different types of information utilized when interpreting an utterance in a natural language interface to a database system. This information corresponds to the information that needs to be analyzed in user-utterances.

Domain concepts are concepts about which the system has information, mainly concepts from the database, T, but also synonyms to such concepts acquired, for instance, from the information base describing the system, S.

In a database query system users also often request information by relating concepts and objects, e.g. *which one is the cheapest*. We call this type of language constructions *relational expressions*. The relational expressions can be identified from the corpus.

Another common type of expressions are numbers. Numbers can occur in various forms, such as dates, object and property values.

Set operations. It is necessary to distinguish utterances such as: *show all cars costing less than 70 000* from *which of these costs less than 70 000*. The former should get all cars costing less than 70 000 whereas the latter should utilize the set of cars recorded as Objects by the dialogue manager. In some cases the user uses expressions such as *remove all cars more expensive than 70 000*, and thus is restricting a set by mentioning the objects that should be removed.

Interactional concepts. This class of concepts consists of words and phrases that concern the interaction such as *Yes*, *No*, etc (cf. (Byron and Heeman, 1997)).

Task/System expressions. Users can use domain concepts such as *explain*, indicating that the domain concept is not referring to a request for information from the database, T, but instead from the system description, S.

When acquiring information for the interpreter, three different sources of information can be utilized: 1) background system information, i.e. the database, T, and the information describing the background system's capabilities, S, 2) information from dialogues collected with users of the system, and 3) common sense and prior knowledge on human-computer interaction and natural language dialogue. The various information sources can be used for different purposes (Jönsson, 1993).

4 The interpretation module

The approach we are investigating relies on analyzing as small and crucial parts of the utterances as possible. One of the key issues is to find these parts. In some cases an analysis could consist of one single domain or interactional concept, but for most cases we need to analyze small sub-phrases of an utterance to get a more reliable analysis. This requires flexibility in processing of the utterances and is a further development of the ideas described in Strömbäck (1994). In this work we have chosen to use PATR-II but in the future constructions from a more expressive formalism such as EFLUF (Strömbäck, 1997) could be needed.

Flexibility in processing is achieved by one extension to ordinary PATR and some additions to a chart parser environment. Our version of PATR allows for a set of unknown words within phrases. This gives general grammar rules, and helps avoiding the analysis to be stuck in case of unknown words. In the chart parsing environment it is possible to define which of the inactive edges that constitute the result.

The grammar is divided into five grammar modules where each module corresponds to some information requested by the dialogue manager. The modules can be used independently from each other.

Domain concepts are captured using two grammar modules. The task of these grammars is to find keywords or sub-phrases in the expressions that correspond to the objects and properties in the database. The properties can be concept keywords or relational expressions containing concept keywords. Numbers are typed according to the property they describe, e.g. *40000* denote a price.

To simplify the grammars we only require that the grammar recognizes all objects and properties mentioned. The results of the analyses are filtered through the heuristics that only the most specific objects are presented to the dialogue manager.

Set operations. This grammar module

provides a marker to tell the dialogue manager what type of set operation the initiative requests, *new*, *old* or *restrict*. The user's utterance is searched for indicators of any of these three set operators. If no indicators are found we will assume that the operator is *old*. The chart is searched for the first and largest phrase that indicates a set operator.

Recognizing interactional utterances. Many interactional utterances are not necessary to interpret for information retrieval systems, such as *Thank you*. However, Yes/No-expressions are important. They can be recognized by looking for one of the keywords *yes* or *no*. One example of this is the utterance *No, just the medium sized cars* as an answer to if the user wants to see all cars in a large table. The Yes/No-grammar can conclude that it is a no answer and the property grammar will recognize the phrase *medium sized cars*.

System/Task recognition. Utterances asking for information about a concept, e.g. *Explain the numbers for rust*, can be distinguished from utterances requesting information acquired from the background system *How rust prone are these cars* by defining keywords with a special meaning, such as *explain*. If any of these keywords are found in an utterance the dialogue manager will interpret the question as system-related. If not it will assume that the question is task-related.

5 An example

To illustrate the behaviour of the system consider an utterance such as *show cars costing less than 100000 crowns*. The word *cars* indicates that the set operator is *new*. The relational expression will be interpreted by the grammar rules:

```
relprop -> property :
           0 properties = 1 properties .

relprop -> property comp glue entity :
           0 properties = 1 properties :
           0 properties = 2 properties :
           0 properties = 4 properties :
           0 properties value arg = 4 value .
```

This results in two analyses [Aspect: price] and [Aspect: price, Value: [Relation: less, Arg: 100000]] which, when filtered by the heuristics, present the latter, the most specific analysis, to the dialogue manager. The dialogue manager inspects the result and as it is a valid database request forwards it to the background system. However, too many objects satisfy the request and the dialogue manager initiates a clarification request to the user to further specify the request. The user responds with *remove audi 1985 and 1988*. The keyword *remove* triggers the set operator *restrict* and the objects are interpreted by the rules:

```
object -> manufacturer :
          0 object = 1 object .

object -> manufacturer * 2 year :
          0 object = 1 object :
          0 object year = 2 year .
```

This results in three objects [Manufacturer: audi], [Manufacturer: audi, Year: 1985] and [Manufacturer: audi, Year: 1988]. When filtered the first interpretation is removed. This is integrated by the dialogue manager to provide a specification on both Objects and Properties which is passed to the background system and a correct response can be provided.

6 Empirical evidence for the approach

In this section we present results on partial interpretation[1] for a natural language interface for the CARS-application; a system for typed interaction to a relational database with information on second hand cars. The corpus contains 300 utterances from 10 dialogues. Five dialogues from the corpus were used when developing the interpretation methods, the *Development set*, and five dialogues were used for evaluation, the *Test set*.

6.1 Results

The lexicon includes information on what type of entity a keyword belongs to, i.e. Objects or Properties. This information is acquired automatically from the database with synonyms added manually from the background system description.

The automatically generated lexicon of concepts consists of 102 entries describing Objects

[1]Results on dialogue management has been presented in Jönsson (1997).

Table 1: Precision and recall for the grammars

	Yes/No	S/T	Set	
Devel. set	100%	100%	97,5%	
Test set	100%	91,7%	86,1%	
	Objects			
	Fully		Partial	
	Recall	Precision	Recall	Precision
Devel. set	100%	98%	100%	98%
Test set	94,1%	80%	100%	85%
	Properties			
	Fully		Partial	
	Recall	Precision	Recall	Precision
Devel. set	97%	97%	99%	100%
Test set	59,6%	73,9%	73,7%	91,3%

and Properties. From the system description information base 23 synonyms to concepts in the database were added to the lexicon. From the *Development set* another 7 synonyms to concepts in the database, 12 relational concepts and 7 markers were added.

The five grammars were developed manually from the *Development set*. The *object grammar* consists of 5 rules and the *property grammar* consists of 21 rules. The grammar used for finding set indicators consists of 13 rules. The *Yes/No grammar* and *System/Task grammar* need no grammar rules. The time for developing these grammars is estimated to a couple of days.

The obtained grammars and the lexicon of totally 151 entries were tested on both the *Development set* and on the five new dialogues in the *Test set*. The results are presented in table 1. In the first half of the table we present the number of utterances where the Yes/No, System/Task and Set parameters were correctly classified. In the second we present recall and precision for Objects and Properties.

We have distinguished fully correct interpretations from partially correct. A partially correct interpretation provides information on the Aspect but might fail to consider Value-restrictions, e.g. provide the Aspect value *price* but not the Value-restriction *cheapest* to an utterance such as *what is the price of the cheapest volvo*. This is because *cheapest* was not in the first five dialogues.

The majority of the problems are due to such missing concepts. We therefore added information from the *Test set*. This set provided another 4 concepts, 2 relational concepts, and 1 marker and led us to believe that we have reached a fairly stable set of concepts. Adding these relational and domain concepts increased the correct recognition of set operations to 95,8%. It also increased the numbers for Properties recall and precision, as seen in table 2. The other results remained unchanged.

Table 2: Precision and recall when concepts from the test set were added

	Properties			
	Fully		Partial	
	Recall	Precision	Recall	Precision
Test set	92,3%	79,5%	93,8%	90,6%

To verify the hypothesis that the concepts are conveyed from the database and a small number of dialogues, we analyzed another 10 dialogues from the same setting but where the users know that a human interprets their utterance. From these ten dialogues only another 3 concepts and 1 relational concept were identified. Furthermore, the concepts are borderline cases, such as mapping the concept *inside measurement* onto the database property *coupé*, which could well result in a system-related answer if not added to the lexicon.

As a comparison to this a traditional non-partial PATR-grammar, developed for good coverage on only one of the dialogues consists of about 200 rules. The lexicon needed to cover all ten dialogues consists of around 470 words, to compare with the 158 of the lexicon used here.

The principles have also been evaluated on a system with information on charter trips to the Greek archipelago, TRAVEL. This corpus contains 540 utterances from 10 dialogues. The information base for TRAVEL consists of texts from travel brochures which contains a lot of information. It includes a total of around 750 different concepts. Testing this lexicon on the corpus of ten dialogues 20 synonyms were found. When tested on a set of ten dialogues collected with users who knew it was a simulation (cf. the CARS corpus) another 10 synonyms were found. Thus 99% of the concepts utilized in this part of the corpus were captured from the information base and the first ten dialogues. This clearly supports the hypothesis that the relevant concepts can be captured from the background system and a fairly small number of dialogues.

For the TRAVEL application we have also estimated how many of the utterances in the corpus that can be analyzed by this model. 90,4% of the utterances can easily be captured by the model. Of the remaining utterances 4,3% are partly outside the task of the system and a standard system message would be a sufficient response. This leaves only 4,8% of the utterances that can not be handled by the approach.

6.2 Discussion

When processing data from the dialogues we have used a system for lexical error recovery, which corrects user mistakes such as misspellings, and segmentation errors. This system utilizes a trained HMM and accounts for most errors (Ingels, 1996). In the results on lexical data presented above we have assumed a system for morphological analysis to handle inflections and compounds.

The approach does not handle anaphora. This can result in erroneous responses, for instance, *Show rust for the mercedes* will interpret *the mercedes* as a new set of cars and the answer will contain all mercedeses not only those in the previous discourse. In the applications studied here this is not a serious problem. However, for other applications it can be important to handle such expressions correctly. One possible solution is to interpret definite form of object descriptions as a marker for an old set.

The application of the method have only utilized information acquired from the database, from information on the system's capabilities and from corpus information. The motivation for this was that we wanted to use unbiased information sources. In practice, however, one would like to augment this with common sense knowledge on human-computer interaction as discussed in Jönsson (1993).

7 Conclusions

We have presented a method for robust interpretation based on a generalization of PATR-II which allows for generalization of grammar rules and partial parsing. This reduces the sizes of the grammar and lexicon which results in reduced development time and faster computation. The lexical entries corresponding to entities about which a user can achieve information is mainly automatically created from the background system. Furthermore, the system will be fairly robust as we can invest time on establishing a knowledge base corresponding to most ways in which a user can express a domain concept.

Acknowledgments

This work results from a number of projects on development of natural language interfaces supported by The Swedish Transport & Communications Research Board (KFB) and the joint Research Program for Language Technology (HSFR/NUTEK). We are indebted to Hanna Benjaminsson and Magne Hansen for work on generating the lexicon and developing the parser.

References

S. Bennacef, H. Bonneau-Maynard, J. L. Gauvin, L. Lamel, and W. Minker. 1994. A spoken language system for information retrieval. In *Proceedings of ICLSP'94*.

Donna K. Byron and Peter A. Heeman. 1997. Discourse marker use in task-oriented spoken dialog. In *Proceedings of Eurospeech'97, Rhodes, Greece*, pages 2223–2226.

Jaime G. Carbonell and Philip J. Hayes. 1987. Robust parsing using multiple construction-specific strategies. In Leonard Bolc, editor, *Natural Language Parsing Systems*, pages 1–32. Springer-Verlag.

Peter Ingels. 1996. Connected text recognition using layered HMMs and token passing. In K. Oflazer and H. Somers, editors, *Proceedings of the Second Conference on New Methods in Language Processing*, pages 121–132, Sept.

Arne Jönsson. 1993. A method for development of dialogue managers for natural language interfaces. In *Proceedings of the Eleventh National Conference of Artificial Intelligence, Washington DC*, pages 190–195.

Arne Jönsson. 1997. A model for habitable and efficient dialogue management for natural language interaction. *Natural Language Engineering*, 3(2/3):103–122.

Kåre Sjölander and Joakim Gustafson. 1997. An integrated system for teaching spoken dialogue systems technology. In *Proceedings of Eurospeech'97, Rhodes, Greece*, pages 1927–1930.

Lena Strömbäck. 1994. Achieving flexibility in unification formalisms. In *Proceedings of 15th Int. Conf. on Computational Linguistics (Coling'94)*, volume II, pages 842–846, August. Kyoto, Japan.

Lena Strömbäck. 1997. EFLUF - an implementation of a flexible unification formalism. In *Proc of ENVGRAM - Computational Environments for Practical Grammar Development, Processing and Integration with other NLP modules.*, July. Madrid, Spain.