# ROBUST INTERPRETATION FOR SPOKEN DIALOGUE SYSTEMS

*Lena Strömbäck and Arne Jönsson*

Department of Computer and Information Science
Linköping University, S-581 83, LINKÖPING, SWEDEN
lestr@ida.liu.se arnjo@ida.liu.se

## ABSTRACT

Spoken dialogue systems must allow for robust and efficient interpretation of user utterances. This can be achieved by using shallow and partial interpretation. Partial interpretation is feasible together with a dialogue manager which provides information to guide the analysis. In this paper we present results on developing interfaces for information retrieval applications utilizing partial and information directed interpretation with unification-based formalisms, traditionally used for deep and complete analysis. The major advantage with our approach is that the time to develop the interpretation modules is reduced. Furthermore, the system will be fairly robust as large parts of the knowledge bases containing knowledge on ways in which a user can express a domain concept can be generated automatically or semi-automatically.

## 1. INTRODUCTION

Traditionally, computational linguistics has developed formalisms for providing deep and complete analysis of natural language. In a spoken dialogue system this 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. Robustness is especially important for spoken interfaces as speech recognition sometimes fails to make a perfect recognition.

When developing knowledge-based spoken dialogue systems for information retrieval applications shallow and partial interpretation are commonly used (cf. [8, 1]). Similar approaches are also proposed in, for instance, the work on flexible parsing [3]. In this paper we will use a unification-based formalism, developed for deep and complete analysis, for shallow and partial interpretation.

Results on shallow and partial parsing with unification-based formalisms for written interaction [6] show that development time and sizes of lexicon and grammar can be reduced. Compared to using unification-based formalisms for deep and complete analysis, the size of the lexicon were reduced to a third and the number of grammar rules reduced from 200 to 39 when tested on 300 user utterances. In this paper we present results from applying the principles for interpretation to spoken interaction. 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 the 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 [5].

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 [5] and interaction information presented as a marker from the interpreter, as discussed below.

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. The distinction between S and T applies to most applications. However, the interpretation might differ. In one of the applications that we have studied, the CARS-application [5], the S knowledge source mainly contains expressions with information on what the system can perform, whereas in another application on bus travel it mainly contains information on where the customer can turn for information not provided by the system such as telephone numbers to lost-and-found departments. In the SUNDIAL dialogues, discussed below, the S knowledge base contains information on for instance other flight companies.

## 3. TYPES OF INFORMATION

The interpreter needs to provide information for the dialogue manager on Objects and Properties, but also additional information. This information corresponds to the information that needs to be analyzed in user-utterances and we identify the following types of information:

**Knowledge base recognition**. In most information retrieval tasks the user can ask questions about different subtasks, such as giving information about a specific flight or providing a telephone number as an answer to a question outside the systems domain. Another example is that 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.

**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 use of relational expressions differs between applications and can be identified from the corpus. Another common type of expressions are numbers. Numbers can occur in various forms, such as dates, and object and property values.

**Set operations**. It is necessary to distinguish utterances such as: *which flight leaves for Paris today* from *which of these arrive before 7 PM.* The former should get all flights arriving to Paris whereas the latter should utilize the set of flights recorded as Objects by the dialogue manager. Users can also use expressions such as *remove morning flights*, to restrict 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. [2]).

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 [4].

## 4. THE INTERPRETATION MODULE

The analysis is done by parsing the parts of the utterances that contain the requested information. The information needed by the interpretation module, i.e. grammar and lexicon, can be derived from the database of the background system and information from dialogues collected in Wizard of Oz-experiments.

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 [9]. In this work we have chosen to use PATR-II [7] which is a well-known example of a unification-based formalism.

Flexibility in processing is achieved by one extension to ordinary PATR and some additions to a chart parser environment. Our version of PATR allows unknown words within phrases which allows for more 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. For the moment we assume a string of words as input for the chart parser, but in the future we could as well allow an n-best lattice from speech recognition.

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 and a description of them is given below.

**Knowledge base recognition**. Utterances asking for information about a concept, e.g. *What does boarding time mean?* or utterances which is partly outside the domain of the system as *Does any other company have flights to Crete today?*, can be distinguished from utterances requesting information acquired from the background system, such as, *When do I have to board?* by defining key-phrases with a special meaning, e.g. *What does* or *other company.* If any of these key-phrases are found in an utterance the dialogue manager will interpret the question as system-related or outside the domain. If not it will assume that the question is task-related.

**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. *NINE PM* denotes a time.

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**. Interactional utterances can be recognized by looking for one of the keywords *yes* or *no*. One example of this is the utterance *No, just BA flights* as an answer to if the user wants to see all flights from a large set. The Yes/No-grammar can conclude that it is a no answer and the property grammar will recognize the phrase *BA flights*. Also, for spoken language systems requests, to repeat the last utterance is quite common and can be recognized by looking for keywords such as *repeat* or *again*.

## 5. AN EXAMPLE

To illustrate the behavior of the system consider an utterance such as *I want to know the arrival time of todays flight from crete*. The relational expression properties in this utterance will be interpreted by the grammar rules:

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

relprop -> direction * 2 time :
           0 properties = 1 properties :
           0 properties = 2 properties .

relprop -> to/from place :
           0 properties = 1 properties :
           0 properties = 2 properties .
```

These rules will result in three analyses [Time: Today], [Time: Today, Type: arrival] and [Place: Crete, Type: departure] which, when filtered by the heuristics, present the two latter, the most specific analyses, to the dialogue manager. The dialogue manager inspects the result and as it is a valid database request can forward it to the background system. In this case there are no British Airways flights to Crete during this day, and the system gives this as a response. The user responds with *is there any other company flying in from crete today*. The keyphrase *other company* triggers the knowledge base marker *give-phone* and Crete again yields an object. On this information the system replies by giving phone-numbers to some other airlines with flights to Crete.

## 6. EMPIRICAL BASE

We have analyzed a corpus collected in Wizard-of-Oz-experiments of 100 information retrieval dialogues, from the SUNDIAL corpus[1]. These dialogues contain more than 300 user utterances. In this application, users request flight information via telephone. A typical interaction from this corpus asks for arrival and departure time of flights to/from Heathrow:

```
S:      flight information british airways good day
        can I help you
U:      #h yes I'm enquiring about flight bee ay
        two eight six , flying in from san francisco
        could you tell me the time of arrival and
        its destination (5)
S:      please wait (18)
        bee ay two eight six is expected at
        thirteen ten (2)
U:      #h (5)
        and where will it arrive (7)
S:      the flight arrives at london heathrow
        terminal four (1.4)
U:      right thank you very much (0.5)
S:      goodbye
```

For the SUNDIAL system we did not have the database and could not derive the lexicon and grammar from it. Instead we used the corpus as the only source, but if the database is available it should form the main source for lexicon development [6].

## 7. RESULTS

One of the aims with this study was to investigate how well the approach works with only a small corpus as base for development for the grammar and lexicon modules. Therefore only the first 60 user utterances were used for grammar development. To this we added information about destinations, airports and other companies from the rest of the corpus, since we assumed that this information would easily be derived from the systems database if it would have been available.

The SUNDIAL dialogues are less complex than the written interaction dialogues analyzed in a previous study [6]. Thus, the grammar modules could be simplified. The most important difference is that no set operation markers are needed. This might be due to the scenario, where the user always asks about one particular flight, as we find the need

[1] These dialogues are from a corpus of Wizard of Oz-dialogues collected and transcribed at the Social and Computer Sciences, University of Surrey, UK as part of the ESPRIT Sundial project (P2218).

Table 1: Precision and recall for the grammars

| | | | | |
|---|---|---|---|---|
| Yes/No | 98,6% | | | |
| Knowledge base | 97,7% | | | |

| | Fully | | Partial | |
|---|---|---|---|---|
| | Recall | Precision | Recall | Precision |
| Objects | 94,3% | 98,8% | 94,3% | 98,8% |
| Properties | 65,4% | 63,5% | 97,1% | 94,3% |

for such markers in another similar corpus of dialogues collected as part of a project on developing a spoken dialogue system for local bus traffic information.

Furthermore, the relational expressions used in the SUNDIAL corpus are much simpler and mainly utilize only a relation restricting a concept, e.g. *before twelve o'clock*, and not comparing objects, e.g. *Are there any cheaper cars* referring to the current set of objects.

The resulting grammar modules contain a total of 39 rules and the corresponding lexicon consists of 105 entries. The development time was approximately ten hours. These grammars and lexicon where tested on the whole corpus of 300 user utterances. The results are presented in table 1. In the first half of the table we present the number of utterances where the Yes/No and Knowledge base parameters were correctly classified. In the second we present recall and precision for objects and properties.

As can be seen from the table we can determine the Yes/No and Knowledge base parameters correct for almost all the cases. Also Objects, i.e the flight mentioned or the requested airline, can be detected correctly in most cases.

For Properties the situation seems worse; only 65% could be fully determined. The most common failure is that the system could not determine if a time or place mentioned by the user denotes an arrival or departure flight. However, for most cases this causes no serious problem, as the system can use other information such as the flight number or the value of other properties to determine the correct interpretation. These cases are referred to as partial recall and the system finds the correct properties in about 97% of the cases.

This means that the system is able to determine all parameters needed by the dialogue manager correctly in 63.3% of the utterances. If we add those utterances where some parameters were only partially correct but where the information still is sufficient for the dialogue manager to provide an answer to the query, the system manages this for 87.2% of the cases. Of the remaining utterances the error is due to a question partially outside the defined scenario in 6.7% of the cases, where a standard system message would be sufficient as an answer. This leaves only 5.8% of the utterances actually being incorrectly handled by the system.

# 8. SUMMARY

In this paper we presented results on shallow and partial interpretation using unification-based formalisms, originally developed for deep an complete interpretation. We identify the types of information needed for dialogue management. Each information type corresponds to a grammar module. The results show that the method give a good recognition of the information concepts and provides a fast and easy way to develop interpreters for spoken dialogue systems.

# 9. ACKNOWLEDGEMENTS

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).

# 10. REFERENCES

1. S. Bennacef, H. Bonneau-Maynard, J. L. Gauvin, L. Lamel, and W. Minker. A spoken language system for information retrieval. In *Proceedings of ICLSP'94*, 1994.
2. Donna K. Byron and Peter A. Heeman. Discourse marker use in task-oriented spoken dialog. In *Proceedings of Eurospeech'97, Rhodes, Greece*, pages 2223–2226, 1997.
3. Jaime G. Carbonell and Philip J. Hayes. Robust parsing using multiple construction-specific strategies. In Leonard Bolc, editor, *Natural Language Parsing Systems*, pages 1–32. Springer-Verlag, 1987.
4. Arne Jönsson. 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, 1993.
5. Arne Jönsson. A model for habitable and efficient dialogue management for natural language interaction. *Natural Language Engineering*, 3(2/3):103–122, 1997.
6. Arne Jönsson and Lena Strömbäck. Robust interaction through partial interpretation and dialogue management. In *Proceedings of Coling/ACL'98, Montréal*, 1998.
7. Stuart M. Shieber. *An Introduction to Unification-Based Approaches to Grammar.* Number 4 in CSLI Lecture Notes. University of Chicago Press, Chicago, Illinois, 1986.
8. Kåre Sjölander and Joakim Gustafson. An integrated system for teaching spoken dialogue systems technology. In *Proceedings of Eurospeech'97, Rhodes, Greece*, pages 1927–1930, 1997.
9. Lena Strömbäck. Achieving flexibility in unification formalisms. In *Proceedings of 15th Int. Conf. on Computational Linguistics (Coling'94)*, volume II, pages 842–846, August 1994. Kyoto, Japan.