LINGUISTIC PHRASE SPOTTING IN A SIMPLE APPLICATION SPOKEN
DIALOGUE SYSTEM

Manuela Boros

Bavarian Research Center for
Knowledge Based Systems (FORWISS)
Am Weichselgarten 7
D-91058 Erlangen, Germany
e-mail: boros@forwiss.uni-erlangen.de

Paul Heisterkamp

Daimler-Chrysler AG
Speech Understanding Systems
Wilhelm-Runge-Str. 11
D-89081 Ulm, Germany
e-mail: paul.heisterkamp@daimlerchrysler.com

ABSTRACT

Spoken dialogue systems have to cope with well known problems of spontaneous speech such as ungrammaticalities, hesitations or corrections. Besides, unrestricted speech poses the problem of paraphrasing, rising the number of utterances exceeding the linguistic coverage of the system. Concerning the linguistic processing of spontaneous speech, partial parsing represents a good method to achieve sufficient robustness against these phenomena. In simple application domains however, mostly systems are implemented that do completely without speech understanding. In our paper we present an approach for robust linguistic speech processing and understanding for simple application domains. Our approach provides a complete analysis of user utterances based on partial analysis of semantically relevant sub-parts of the utterance. The semantic representations may be directly mapped onto data base indices thus bypassing the interpretation by the dialogue manager. On the other hand, the integration into complex dialogue systems allows for free dialogues and recovery strategies in case of system faults and misunderstandings.

1. INTRODUCTION

Simple application domains for speech dialogue systems may be characterized by three main features: first, dialogues are rather short, as information requests may be formulated easily in just one sentence, second, typical user requests follow only a small number of different syntactic templates, and third, the information requested can be stored in a data base and easily be retrieved from there without any complex inference mechanisms. Dialogue systems for simple applications usually are realized in terms of simple approaches such as keyword spotting and strictly menu driven IVR systems. In these cases, dialogue is severely restricted: the user must utter (only) the keywords covered by the system and follow its instructions carefully because the system only accepts the requested user reactions. This inflexibility is due to the lack of interpretation of user utterances that help to detect the user’s intention. More sophisticated dialogue approaches, on the other hand, allow for rather flexible and free dialogues using unrestricted spontaneous speech. Up to now, such systems have been employed only in applications, that are enough complex to demand complex systems. If applied to simpler applications, these dialogue systems probably would produce a large overhead of functions that are not necessary in the specific domain. Also, unrestricted speech poses new problems to word recognizer and linguistic processor. This is due to well known phenomena like ellipses, hesitations, corrections, ungrammaticalities and out-of-vocabulary words. From the linguistic point of view, paraphrasing also is a problem that easily exceeds the boundaries of the system’s coverage. As an example consider the German word Lichtschalter (light switch) that can be paraphrased by the noun phrase Schalter für das Licht (switch for the light). Covering paraphrases in the system’s grammar leads to a rise in complexity and size, and thus in development costs, especially in free word order languages, where all the combinators of the paraphrasing expressions have to be taken into account. Due to these problems, simple application domains are most often addressed by simple keyword spotting approaches as described above.

We present an approach to partial analysis in spoken dialogue systems, that is based on the notion of phrase spotting. Phrase spotting, a special kind of partial parsing, as opposed to keyword spotting, means that the parser will search for special syntactic or semantic parts (phrases) of a sentence, that will be analyzed and assigned a semantic representation for further interpretation. Our approach differs from other approaches to partial parsing in dialogue systems in two ways: first, even though we analyze only parts of an utterance, the parser output will consist of one single semantic representation for the contents of that utterance, and second, these semantic representations will be matched directly onto the database indices without further interpretation through the dialogue manager. This means that the complex dialogue strategies only have to be invoked in case of a mismatch between semantic representation and data base indices, which is the case for misunderstandings or underspecified requests made by the user. Our approach therefore allows simple application domains to benefit from the advantages of complex systems strategies without making things unnecessarily complicated.

In the following we will provide a closer description of the phrase spotting approach and the application dependent lexicon and grammar, cf. section 2. The integration of the phrase spotter into the dialogue system and preliminary results will be shown in section

ISCA Archive
http://www.isca-speech.org/archive

6th European Conference on
Speech Communication and Technology
(EUROSPEECH’99)
Budapest, Hungary, September 5-9, 1999
3. The paper ends with a short conclusion and an outlook on further work in section 4.

2. THE PHRASE SPOTTING APPROACH

Partial parsing techniques are commonly used to cope with spontaneous speech or recognizer faults in spoken dialogue systems, as described in [1] and [2]. In these approaches, the parser searches the recognizer output for meaningful fragments that satisfy the given grammar. In [3] a keyword spotting approach is described, that also applies a kind of partial parser that tries to find meaningful sequences of keyword from a keyword lattice. The phrase spotting approach presented throughout this paper is situated between these two approaches. In the spoken dialogue system, the phrase spotter is the core of the linguistic processor. It takes as input graphs of word hypotheses, as generated by the word recognizer, parses them and provides a semantic representation of the underlying utterance. The analysis is driven with the aid of a linguistic knowledge base, comprising the lexicon and grammar rules for the respective domain. Figure 1 shows the architecture of the linguistic processor.

![Figure 1. The system’s linguistic processor.](image)

2.1. The Basic Idea

Like any speech parser, the phrase spotter searches the word graph for the best scored and grammatically correct path. Grammatical correctness is defined by the system’s grammar. In the phrase spotting approach, the grammar is restricted to cover only parts of user utterances, thus resulting in a very fragmented grammar. The parts of an utterance, to be covered by the grammar, are defined in terms of relevant parts. By relevant parts we mean those parts, that are crucial for its correct interpretation in the current domain. For clarification consider the sentence *Where is the light?*. For the correct interpretation of this sentence in the domain we addressed, only the interrogative *where* and the noun phrase *the light* are relevant (cf. section 3). The verb *to be* may be neglected as it is – in the addressed domain – implicitly given by the interrogative *where*. This restriction in coverage of the linguistic knowledge results in the drastic reduction of the knowledge base’s size, and therefore allows for the additional coverage of paraphrasing expressions without exceeding a critical point in complexity. Also development costs are kept quite low.

So far, our approach follows the traditional approach of partial parsing. As an extension to that, the phrase spotter not only generates partial analyses for each utterance, but also combines the analyzed parts to one single result. This combination even takes place across gaps in the graph that are due to neglected or not covered words or phrases. In the above given sentence *Where is the light?* the phrase spotter would thus analyze the relevant parts where and the light and combine the two to one single result *where the light*. The idea behind this step is to achieve one single semantic representation instead of several partial representations. In simple information retrieval domains, these semantic representations may be used directly as data base indices to retrieve the requested information. In consequence, the linguistic knowledge base will contain not only the rules to analyze the relevant parts of an utterance but also rules to combine these parts into complete parses. These additional rules are called meta rules as they do not necessarily correspond to the traditional grammar theory of natural language, c.f. section 2.2.

2.2. The Spotting Algorithm

The phrase spotter is implemented in terms of an agenda-driven island-based active chart parser, such as described in [4]. It tries to find a complete path through its input graph and outputs the corresponding semantic representation. Complete means, that even though the phrase spotter only looks for the relevant parts of the graph, it will generate a full spanning edge and thus one single semantic representation for the utterance. Therefore, linguistic phrase spotting is divided into two different levels of parsing, that will be explained in the following. To keep things simple, we will use word chains instead of word graphs in our example, the real phrase spotter, however, is operating on word graphs.

2.2.1. Parsing Level 1

At the beginning of the parsing process, the chart is initialized with the lexical entries corresponding to the word hypothesis in the input word graph. In order to avoid gaps in the chart, that occur if a word hypothesis cannot be found in the lexicon, for each hypothesis not covered by the lexicon, a dummy edge is inserted into the chart. Dummy edges do not have any linguistic properties such as syntactic or semantic constraints, but they keep the acoustic score of their original word hypothesis as well as the information of start and end nodes in the graph. Dummy edges are important in order of keep the original structure of the graph with all possible paths. Figure 2 shows the initial chart for the input sentence *Wo ist der Lichtschalter?* (Where is the light switch?) including dummy edges.

![Figure 2. Initial chart](image)

During parsing level 1 dummy edges are not being considered for combination with adjacent edges. Therefore, the parser only expands regular chart edges until no further expansion can take place. The
island parsing approach guarantees that all possible maximal chart edges can be generated, as it allows the parsing process to start at arbitrary points (islands) in the graph. In our case, parsing starts with the best scored word hypotheses that are expanded successively to the left and right. Parsing level 1 terminates if either a full spanning edge has been generated or the regular chart edges cannot be further combined because of adjacent dummy edges or no fitting grammar rules. Figure 3 shows the chart for the current example after termination of parsing level 1.

Figure 3. Chart after parsing level 1.

After applying the grammar rules during parsing level 1, the noun phrase der Lichtschalter has been generated as new chart edge. The chart now contains two maximal chart edges that cannot be further expanded because of the dummy edge separating them. Each maximal chart edge in the chart, after parsing level 1 has been completed, represents one relevant part of the utterance, as defined above.

Full spanning edges after parsing level 1 occur more often than one would intuitively think. Especially in the addressed simple domains, utterances comprising just one single word or one single relevant expression are quite common, as Wizard of Oz experiments showed (cf. section 3).

2.2.2. Parsing Level 2

In case that parsing level 1 does not provide a full spanning edge, i.e., a complete parse of the word graph, the phrase spotting will run parsing level 2, where also dummy edges are processed. During parsing level 2, chart edges may be combined across dummy edges, applying the meta rules in the grammar in order to combine the relevant parts identified during level 1. Due to the dummy edges, it is made sure, that chart edges may only be combined if they are part of the same path through the original graph. Figure 4 shows the chart after applying meta rules and parsing across dummy edges.

Figure 4. Chart after parsing level 2.

As can be seen in figure 4 the result of the phrase spotting process is a full spanning edge rather than several partial edges, as in most other partial parsing approaches. Due to the meta grammar rules, also one single semantic representation for that full spanning edge has been generated that is passed directly to the database to check, whether it represents a valid database index.

Figure 4 also makes clear that chart edges, resulting from applying the meta grammar rules, do not necessarily represent correct sentences in the sense of natural language grammar. It is obvious, that the combination of partial analyses of sentences, while leaving syntactically important parts like verbs, won’t lead to grammatically correct results. However, the goal of the parsing process is not to provide a correct syntactic or semantic analysis of the input, but to generate a semantic representation that allows for its correct interpretation in the context of the application domain.

3. SYSTEM INTEGRATION

The phrase spotter was integrated into a state-of-the-art spoken dialogue system replacing the original linguistic processor. The system was adapted to the domain of operating instructions for cars accessible via spoken language: The user may ask the system for information on various topics from the instructions while sitting in the car.

In order to provide a useful system, the original paper version of the operating instructions had to be restructured. 289 significant different topics have been identified and stored as audio data into a data base. For each topic in the instructions there are at most three types of information: definitions, operating instructions and warnings, leading to around 800 data base entries.

3.1. Wizard of Oz Experiments

While simulating the system in Wizard-of-Oz experiments, typical dialogues with naive users have been collected. The evaluation of these dialogues revealed three interesting facts. First, a rather small number of different syntactic patterns can be identified, that are used by almost every user to formulate his requests. Second, once a pattern lead to success, i.e., the requested information was provided to the user, he would never change his strategy again until a misunderstanding occurs. Third, the user utterances are quite short, in many cases, the user will just say the name of the topic he wants to know something about. These observations allowed for the identification of those syntactic patterns, that the system must cover in order to understand the user’s queries. For each type of information, provided for a topic, table 1 shows the most common syntactic patterns of their realization.

<table>
<thead>
<tr>
<th>Type</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>Was ist <strong>THEME</strong>? (What is <strong>THEME</strong>?)</td>
</tr>
<tr>
<td>Operating</td>
<td>Wie funktioniert <strong>THEME</strong>? (How works <strong>THEME</strong>?)</td>
</tr>
<tr>
<td>Instructions</td>
<td></td>
</tr>
<tr>
<td>Warnings</td>
<td>Was muß ich bei <strong>THEME</strong> beachten? (What do I have to mind with <strong>THEME</strong>?)</td>
</tr>
</tbody>
</table>

Table 1. Patterns for information request.
**THEME** stands for any expression that is associated with one of the topics in the instructions. This may be a single noun, as well as complex phrases (e.g. paraphrasing expressions for nouns). In many cases, in which the user asked for the definition of something, he only uttered the topic name. Questions concerning the operating instructions may as well be formulated using another verb than *function() (works)*, in any case, however, the interrogative *what* (how) and the Theme has to be mentioned.

Table 1 shows, that each information request may be unambiguously assigned the type of information that is asked for, only by looking at its syntactic pattern. Accordingly, the semantic representation for each request will mirror the type of information.

### 3.2. Grammar Development

The syntactic coverage of the grammar fragment was determined on the basis of the syntactic patterns in table 1. For most patterns, the verbs can be neglected because the type of requested information can be inferred from the interrogative directly. The only exception is made for the type of warnings: like for definitions, the interrogative used here is *was* (what), so the verb has to be taken into account to get the difference. Besides, also the different topics have to be covered by the grammar. In many cases, topics are referred to by simple or compound nouns, but also complex noun phrases occur quite often, either as paraphrases for compound nouns or as original topic name. Table 2 shows some of the typical complex noun phrases paraphrasing or representing original topics.

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Nouns</td>
<td>Light, Gang, Tank</td>
</tr>
<tr>
<td>Compound Nouns</td>
<td>Lichtschalter, Tankanzeige</td>
</tr>
<tr>
<td>Noun Phrases</td>
<td>Lösen der Handbremse</td>
</tr>
<tr>
<td></td>
<td>position of the steering wheel</td>
</tr>
</tbody>
</table>

Table 2. Examples for complex noun phrases.

For each topic a unique semantic representation is chosen that also serves as data base index. Also, each type of information is assigned its own semantic representation, so that different types of information can directly identified by their semantics. The current grammar, covering most of the themes and their paraphrases, comprises 12 grammar rules, only 4 of which are meta rules. The lexicon is quite large with 584 entries, most of them nouns.

### 3.3. System functionality

In all cases in which the user will formulate his request in one utterance that allows for the generation of a complete semantic representation and direct data base access, the phrase spotting approach will work like a keyword spotter as the answer is directly output to the user without further involvement of the dialogue manager. In cases of underspecified user requests or severe recognition errors, it is of great advantage employ dialogue strategies that allow for clarification sub-dialogues. Consider the following example dialogue:

**User:** Well, where is this light switch now?
**System:** Do you mean the switch for the headlights or for the interior light?
**User:** For the interior light.
**System:** The switch for the interior light is above the windscreen next to the internal mirror.

The user provides incomplete information on the requested topic; there are a lot of different lights in a car. Instead of failing or choosing more or less randomly a certain number of topics fitting the keyword *light*, the system asks the user to further specify the topic. This is done in a very simple way: if the semantic representation generated by the phrase spotter does not match a data base index, this representation is passed on to the dialogue manager for evaluation. The dialogue manager's task is to drive the dialogue as far, as a complete semantic representation matching a data base index has been provided.

### 4. CONCLUSION AND FURTHER WORK

We presented a robust and efficient approach for linguistic phrase spotting in spoken dialogue system. The phrase spotter differs from other partial parsing methods in its capability to combine partial analyses across gaps in the chart. Two major advantages can be drawn from this approach: First, the underlying grammar can be kept small and cheap in its development costs. Second, the spotter can easily be integrated into a state-of-the-art dialogue system, allowing to reduce complexity by directly mapping semantic representations to data base indices.

In the future, we plan to run field tests with the integrated system, in order to optimize phrase spotter and the dialogue strategies.

### REFERENCES


