# How Statistics and Prosody can guide a Chunky Parser

Manuela Boros[1], Jürgen Haas[2], Elmar Nöth[2], Volker Warnke[2], Heinrich Niemann[2]

[1] Bavarian Research Center for Knowledge Based Systems (FORWISS)
Am Weichselgarten 7
D-91058 Erlangen
Germany
email: boros@forwiss.uni-erlangen.de
[2] University of Erlangen-Nürnberg
Chair for Pattern Recognition
Martensstraße 3
D-91058 Erlangen
Germany
email: (haas,noeth,warnke,niemann)@informatik.uni-erlangen.de

## 1 Introduction

Following the most common architecture of spoken dialog systems as shown in Figure 1, the main task of linguistic processing is to yield a semantic representation of what the user said.

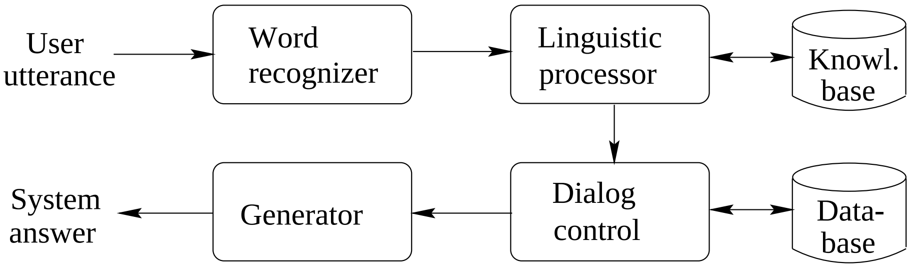

**Figure 1.** Typical dialog system architecture.

These semantic representations are interpreted by the dialog module according to the dialog context and the system answer will be generated accordingly. The system utterance depends on whether the system still needs certain information or if all necessary information has been given to accomplish its task. In order to know, when all required information has been provided, the dialog

This work was partly funded by the European Community in the framework of the SQEL–Project (Spoken Queries in European Languages), Copernicus Project No. 1634. The responsibility for the contents lies with the authors.

module needs to keep track of the relevant semantic concepts uttered by the user. Relevance of semantic concepts is predefined according to the respective, most often closely restricted domain.

Linguistic processing in a dialog system has to cope with the problem of unrestricted (spoken) text, that most often is ungrammatical e.g. contains hesitations, repetitions, ellipses or corrections. However, to yield a successful dialog in a restricted domain only the semantically relevant parts of user utterances are important, e.g. only those concepts, that can/must be interpreted by the dialog manager. Therefore, it seems to be legal to restrict linguistic analysis to the semantically relevant parts and to skip over the rest. Figure 2 shows an example for the attachment of concepts to the appropriate parts of an utterance.

| | | |
|---|---|---|
| Hello, I'd like to go | to Munich | tomorrow. |
| | goal | date |

**Figure 2.** Relevant concepts in an utterance.

For example, in the case of a train timetable information system, values for the semantic slots *time, goal, source*, and *date* have to be filled in order to perform the database query. We therefore suggest that linguistic analysis should be restricted to these concepts, resulting in several grammar fragments that cover the concepts' syntactic surface forms. As a result, the system's linguistic database (grammar) will be much smaller and more modular, allowing for easy maintenance and reuse in other domains. Since parts of the utterances can be neglected during parsing, the processing time is expected to be much shorter than for a full covering grammar. The partial concept grammars can be developed by using corpora of examples of surface structures for the respective concept. In section 2 we will describe the used grammar formalism and first experiments.

### 1.1 Introduction of Statistics and Prosody

When restricting the linguistic analysis to few surface forms or semantic concepts, still one question remains to be answered: which concepts does one have to look for in a given utterance, e.g. which grammar fragment does the parser have to use for each turn? Most often the expected concepts can be predicted by the dialog system due to the last system utterance. For example if the system asked *At what time do you want to leave?* the answer is expected to contain a time expression. However, the user not necessarily answers with specifying a time of departure. He also may correct misunderstandings of information given before or provide additional information like where he wants to leave from, as the following examples show:

```
system: How may I help you?
user:   I want to go to Trier.
system: When do you want to leave
        to Kiel?
user:   I want to go to Trier.
```

```
system: How may I help you?
user:   I want to go to Munich
system: When do you want to leave?
user:   Tomorrow at six from Berlin.
```

In such cases, a device seems to be useful, that is able to tell the parser which concepts it has to look for in a given utterance. In order to provide this kind of information we use statistical methods that take the recognizer output as input and compute the probabilities of each concept to occur in that utterance. This kind of predictor is described in section 3.

However, even if the parser knows, which concepts it has to look for, i.e. which grammar fragment to apply, when dealing with word hypotheses graphs it may find several competing concept tokens. In order to increase the probability of detecting the correct user utterance, word hypotheses graphs may contain more than one path, each of them representing one potential user utterance. The parser, e.g. when looking for time expressions, has to search the complete graph to get all alternatives, and must decide which of them to choose as the correct (i.e. the one that was actually uttered) to be passed on to the dialog module. Usually, the choice among competing alternatives is driven by the usage of acoustic scores as delivered by the word recognizer. However, in order to be able to detect the correct (partial) path through the graph as quickly as possible, analysis should be started at the correct alternative already. To make this possible, we use prosodic information that provides plausibility scores, which are merged with the acoustic scores, thus improving scoring of the hypotheses. Computation of prosodic scores is described in section 4.

The resulting system, integrating partial parsing, statistic concept prediction and prosody, is expected to work very efficiently on either word strings or lattices and can replace the full linguistic analysis as used in our system so far.

## 2 The Chunky Parser

The parser's task is to locate and semantically analyze those parts of a user utterance that have to be interpreted by the dialog module. In this sense, it behaves like a chunk parser, whose aim is to find all the chunks in a given sentence without necessarily attaching them to a complete analysis. The main differences between a chunk parser and our chunky parser is the fact, that in our case just the relevant parts are located by ignoring the rest of the sentence and that the *chunks* we look for are motivated semantically only.

The parser of the system's linguistic processor is an unification based island driven chart parser, that can handle either word strings or word hypotheses graphs. In case of word graphs, the island parsing strategy is applied by starting the analysis at several points in the graph, trying to expand these *islands* successively to the left and to the right. In either way, the parser may start at an arbitrary point in the recognizer output and expand the analysis in both directions. For the detection and analysis of semantic chunks, the possible surface forms are described by a grammar fragment per semantic concept. The linguistic knowledge base therefore does not contain one full grammar, but several grammar modules, that may be grouped individually together if the application changes.

The grammars for the chunky parsers are developed in terms of a context-free phrase structure grammar that comprises lexica and bundles of context-free grammar rules. Each linguistic sign (lexical entries as well as complex expressions) is described through a complex feature structure, whose main features cover morphologic, syntactic, and semantic information. As the aim of linguistic processing in a dialog system is to yield a semantic representation of user utterances, most emphasis lies on its semantic representation in the feature structure. Semantic representation is done on the basis of a semantic inheritance network in which world knowledge can be modeled. Each word or expression is given a semantic type, that can be further specified by so-called roles. By this way, also syntactic dependencies can be mirrored in the network, e.g. the dependencies between verbs and their object. As an example the semantic representation of the sentence *I want to go to Munich.* is given in figure 3.

$$\begin{bmatrix} type : go & \\ thegoal : & \begin{bmatrix} type : location & \\ thecity : & \begin{bmatrix} type : city \\ value : munich \end{bmatrix} \end{bmatrix} \\ thetheme : & \begin{bmatrix} type : individual \\ thediscourserole : \\ \qquad\qquad \begin{bmatrix} val : speaker \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

**Figure 3.** Full semantic representation.

However, the chunky parser only searches for and analyses semantically relevant parts of an utterance, e.g. the part *to Munich* as realization of the concept *goal*, the resulting representation of *I want to go to Munich* looks like the one given in figure 4.

$$\begin{bmatrix} type : location & \\ thecity : & \begin{bmatrix} type : city \\ value : munich \end{bmatrix} \end{bmatrix}$$

**Figure 4.** Partial semantic representation.

### 2.1 Adding Statistics and Prosody

Without any additional knowledge, the parser does not know which concepts it should look for in an utterance, i.e. which grammar to apply on the input. As a result, it will try each grammar fragment of the knowledge base, which would almost be the same as to parse with a full grammar. We therefore use a statistically trained predictor (see section 3), that provides the parser with information on which kind of concept it has to look for in the utterance. The parser then only activates the relevant grammars and fails in case none of them can be applied to the utterance, i.e. none of the predicted concepts can be found.

In order to guide the search for chunks by the parser, prosodic information (see section 4) is merged into the acoustic scores delivered by the recognizer. Prosodic scoring results in better scores for words that belong to one of the relevant concepts that are included in the utterance. Thus, the parser will start with the best scored word, trying to extend it to the left and to the right according to the chosen grammar fragment. This strategy is expected to render the parsing process more efficiently.

## 2.2 Experiments

The first grammar fragment we developed for the chunky parser covers time expressions, i.e. the semantic concepts `date` and `time` that we merged for the first prototype. Typical time expressions that occur in a train timetable information system are hours (e.g. *at nine o'clock*), dates (e.g. *on the first of July*), weekdays (e.g. *on Monday*), or relative dates (e.g. *tomorrow*).

Grammar fragments are developed by collecting sample data (e.g. by WOZ experiments) and choosing some representative examples from it. Based on this, a first grammar version is defined and tested and extended successively.

First experiments were run on a corpus of utterances that were collected while our system was available over the public telephone network. A set of 119 utterances was chosen from the corpus covering only time expressions. These utterances contain 389 words which yield 2618 word hypotheses in the resulting graphs. Semantic accuracy was measured using the metric of semantic concept accuracy (CA) following [1]. For comparison we also parsed these utterances with the full grammar (unification categorial grammar) for train timetable information used so far in the system. We intend to completely replace the UCG by grammar fragments in the near future. Experiments were run on the word hypotheses graphs as well as on the transcriptions of the user utterances (thus simulating 100 % word recognition). Table 1 shows the resulting figures; results on the transcriptions are marked with *script*, results on word graphs are marked with *graph*.

| | | |
|---|---|---|
| # of utterances | 119 | |
| # of hypotheses | 2618 | |
| | Full Grammar | Fragment |
| time script | 9.2 s | 4.6 s |
| CA script | 98 % % | 89 % |
| time graph | 67.5 s | 18.0 s |
| CA graph | 59 % | 68 % |

**Table 1.** First experiments with a Grammar Fragment.

The resulting figures for CA on transcriptions show, that the semantic coverage of the full grammar is very good, whereas the time grammar does not yet achieve the same coverage. CA on the word graphs decreases for the full grammar more than for the fragment. Also the processing time is much higher for the full grammar especially for word graphs. The reason for these observations lies in the fact that the chunky parser using a grammar fragment just searches the hypotheses graph for paths belonging to the expected semantic concept. The full grammar, on the other hand, has to search the full graph to find a complete path through it, which also increases the possibility of following the wrong path resulting in a loss in CA.

## 3 The Semantic Concept Predictor

We examine a statistical approach using $n$-gram language models as semantic concept predictor. The model has to decide about the occurrence of special semantic concepts in word chains. We prove its usability on a corpus collected with the above mentioned information retrieval system containing the 119 utterances for the grammar development. The predictor should be able to decide whether there appears a time expression in an utterance or not. The method then can be extended to other semantic concepts like *goal, source*, and *date.*

### 3.1 Language Model Predictor

The language model we use computes estimations for the occurrence of a word $w_i$ under the assumption of its predecessor words $w_{i-1}, \ldots, w_{i-n+1}$ as the linear interpolation of the $n$-gram and all smaller $n$-grams to smooth the probabilities. Denoting the number of words in the lexicon by $L$, the factor $\frac{1}{L}$ describes the so called zerogram. The interpolation weights $\rho_i$ are used as weighting factors for the linear interpolation and they are estimated automatically with a cross validation technique [3]. The probability $p(\underline{w})$ for a word chain $w_1 \ldots w_m$ is given by the following equation:

$$p(\underline{w}) = \prod_{k=1}^{m} \left( \rho_0 \cdot \frac{1}{L} + \rho_1 \cdot p(w_k) + \sum_{i=2}^{k} \rho_i \cdot p(w_k \mid w_{k-i+1} \ldots w_{k-1}) \right) \quad (1)$$

If we now want to use $n$-gram language models as a semantic concept predictor we have to claim for a word chain $\underline{w}$ whether the concept we are looking for is expressed in $\underline{w}$ or not. For this purpose we build two different language models. The first one is trained with word chains expressing the semantic concept and the second one with the utterances not giving this special concept. During analysis we compute the two scores for the incoming word chain – when using word graphs we choose the best word chain in the graph – and we decide for the higher probability.

A more detailed analysis is possible when we split the training data into three different sets, which are used to train language models. The first set comprises all word chains where the semantic concept does not appear, the second one all utterances where only the concept is expressed and no other semantically relevant information is present and the third set all word chains where the interesting semantic concept appears along with additional information. The decision rule again decides for the highest probability of the three scores. If we combine the second and the third part we have a probability estimation for the same partition as above.

### 3.2 Experiments

In the experiments we show that the $n$-gram language models are able to be used as predictors for semantic concepts. For training and test purposes we use 10114 sentences collected with the above mentioned system. Here we concentrate on the detection of time expressions in the examined word chains. Therefore we mark, based on the transliteration of the utterance, each of the 10114 sentences whether there is the semantic concept *time* or not (NO), and if it is there we mark whether there appears only this time expression (ONLY) or even more task relevant information (PLUS).

The available data is split 2/3 to 1/3 for training and test purposes. The number of sentences for each class is presented in Table 2. Since a word sequence from the test set might have been used by a different speaker from the training set (i.e. if the system asks *where do you want to leave from* different users answered with the same city name expression), the column 'test ≠ train' gives the number of sentences from the column 'test' that were not observed during training.

|  |  |  | test ≠ train |  |
|---|---|---|---|---|
|  | train | test | spoken | recog. |
| ONLY | 92 | 45 | 16 | 17 |
| PLUS | 1070 | 535 | 313 | 375 |
| NO | 5582 | 2790 | 874 | 1227 |

**Table 2.** Number of word chains for training and testing

We train 5-grams for the three different classes and use linear interpolation of the scores for different length of the history. The 'Semantic Concept Predictor' results we obtain are reported in Table 3.

We see that our language model approach to the prediction task performs quite well and could therefore be used as a predictor for the semantic concept analysis. For the two class problem, where we only want to decide whether a time expression occurs or not we obtain a recall of 97.6% and a precision of 54.9% for the spoken word chain and 72.4% recall and 49.0% precision for the recognized.

|  | spoken word chain |  |  |  | recognized word chain |  |  |  |
|---|---|---|---|---|---|---|---|---|
|  | ONLY | PLUS | NO | RR | ONLY | PLUS | NO | RR |
| ONLY | **91.1** | 8.9 | 0.0 | **85.3** | **71.1** | 11.1 | 17.8 | **81.1** |
| PLUS | 8.8 | **88.8** | 2.4 |  | 6.0 | **65.6** | 28.4 |  |
| NO | 7.2 | 8.2 | **84.6** |  | 6.9 | 8.8 | **84.3** |  |

**Table 3.** Detection rates with the spoken (left) and recognized (right) word chain

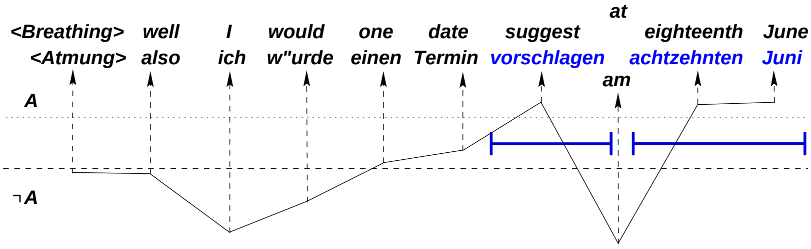

**Figure 5.** A German sentence from Verbmobil-database with probability $P(A \mid w)$ for each word $w$ and the two estimated focused regions with word to word translation.

## 4 Prosodic Scoring

The use of prosodic information for spoken dialog systems becomes more and more important. In the Verbmobil-project [2] prosodic information was successfully used in a speech understanding systems for the first time.

Using a neuronal network (NN) with different sets of prosodic features like pitch-contour, energy-contour and duration of words, syllables and syllables nucleus, phrase boundaries, sentence mood and phrase accent are determined [6].

We want to use this prosodic information to determine the focused regions in a phrase. These regions are the parts of a sentence, which hold the most important content words e.g. time expressions and locations. To get information for the focused regions, we use a NN trained on a part of the Verbmobil-database with a topology of 276 nodes in the input-layer (one node for each used prosodic feature), a hidden-layer with 60 nodes and an output-layer with 2 nodes (a word is accentuated $\mathsf{A}$ or not $\neg\mathsf{A}$). Using the scores $Score(A \mid w)$ and $Score(\neg A \mid w)$ from the output-nodes of the NN for each word $w$ we can estimate the probability $P(A \mid w)$ by using the simple formula

$$P(\mathsf{A} \mid w) = \frac{Score(\mathsf{A} \mid w)}{Score(A \mid w) + Score(\neg A \mid w)}.$$

Now we are able to estimate the probability $P(A \mid w)$ for each word of an utterance and can decide for a focused region by using a threshold. In Figure 5 an example is given for a German utterance.

If we are now able to estimate the stressed regions in a given utterance there are two possible ways to use this knowledge in combination with our parser:

1. we rank the regions by their prosodic scores and offer the ranking list to the parser, which has to find the best expression for the given context
2. we get a list of possible expressions from the parser and disambiguate them using the prosodic score from the NN

Both ways can efficiently be used to find the best expression the parser is searching for in the context the concept predictor has estimated. The first way may be the better one if we are working on word hypotheses graphs, because the parser only has to search in the best scored paths and the search effort may be smaller.

### 4.1 Experiments

In this section we present results we achieved for determining stressed words for different dialog acts (see [5]) in the VERBMOBIL-database using the above described NN. In VERBMOBIL there are 42 illucotionary dialog acts defined which can be grouped into 18 classes. For these 18 dialog act classes, which are used for template based translation, we estimated the most frequent stressed words of a subset of the VERBMOBIL-database using the above described method. For this approach we only ranked those words, whose stress probability exceeds a threshold of 0.8 and were seen stressed in more than 80% of their occurrences. In Table 4 one can see the ten most often seen automatically estimated stressed words for all dialog act classes together. Table 5 shows the five most often seen detected stressed words for the most frequent dialog act classes SUGGEST and ACCEPT. In both tables the words are ranked by their frequency of occurrence in the observed data set.

| $P(A \mid w) > 0.8$ | | |
|---|---|---|
| *Rank* | *% stressed* | *word (translation)* |
| 1 | 88.57 | Freitag (Friday) |
| 2 | 82.69 | Wiederhören (bye) |
| 3 | 84.31 | Donnerstag (Thursday) |
| 4 | 90.91 | Samstag (Saturday) |
| 5 | 95.35 | neunzehnten (19th) |
| 6 | 81.82 | August (August) |
| 7 | 96.15 | vierundzwanzig. (24th) |
| 8 | 87.50 | achten (8th) |
| 9 | 86.96 | wunderbar (marvellous) |
| 10 | 100.00 | sechsundzwanzig. (26th) |

**Table 4.** Automatically determined stressed words for all dialog acts.

| | ACCEPT | | SUGGEST | |
|---|---|---|---|---|
| | $P(A \mid w) > 0.8$ | | $P(A \mid w) > 0.8$ | |
| *Rank* | *% stressed* | *word (translation)* | *% stressed* | *word (translation)* |
| 1 | 100.00 | einverstanden (ok) | 82.22 | Montag (Monday) |
| 2 | 100.00 | Ordnung (alright) | 87.80 | Freitag (Friday) |
| 3 | 100.00 | wunderbar (marvellous) | 83.33 | Donnerstag (Thursday) |
| 4 | 85.71 | Freitag (Friday) | 82.76 | Mittwoch (Wednesday) |
| 5 | 85.71 | frei (free) | 93.10 | Samstag (Saturday) |

**Table 5.** Automatically determined stressed words for dialog act ACCEPT and SUGGEST.

The results from tables 4 and 5 show, that we are able to detect the content words of an utterance if we determine the stressed words. This fact is very important for the use of this method to estimate the focused regions by only using acoustic features to decide for semantically important information.

## 5 Conclusion and Further Work

In our paper we presented a concept for partial semantic parsing in a spoken dialog system. Syntactic analysis is restricted to those parts of user utterances that contain semantic concepts necessary for understanding. The parser acts like a chunk parser in that it looks for subparts in the input string. In our approach, these subparts correspond to semantic concepts and are not necessarily adjacent to each other, i.e. parts of the sentence may stay unreviewed. Syntactic analysis is guided by grammar fragments that are activated by a semantic concept predictor that can tell which concepts occur in a given utterance. This information is supported by prosodic scores that assign higher probability to words that belong to predicted concepts. Each of the subparts has been tested in preliminary experiments that show encouraging results.

Next we plan to extend the number of grammar fragments and to integrate the system in order to verify our assumptions on accuracy and efficiency.

## References

1. M. Boros, W. Eckert, F. Gallwitz, G. Görz, G. Hanrieder, and H. Niemann. Towards understanding spontaneous speech: Word accuracy vs. Concept accuracy. In ICSLP 96 [4], pages 1005–1008.
2. T. Bub and J. Schwinn. Verbmobil: The Evolution of a Complex Large Speech-to-Speech Translation System. In ICSLP 96 [4], pages 1026–1029.
3. F. W. H. Ney, S. Martin. Statistical language modeling using leaving-one-out. In G. B. S. Young, editor, *Corpus-based Methods in Language and Speech Precessing*, pages 210–234. Kluwer Academic Publishers, Boston, 1997.
4. *Int. Conf. on Spoken Language Processing*, Philadelphia, Oct. 1996.

5. S. Jekat, A. Klein, E. Maier, I. Maleck, M. Mast, and J. Quantz. Dialogue Acts in Verbmobil. Verbmobil Report 65, April 1995.
6. A. Kießling. *Extraktion und Klassifikation prosodischer Merkmale in der automatischen Sprachverarbeitung.* Berichte aus der Informatik. Shaker Verlag, Aachen, 1997.