ABSTRACT

Parsing can be improved in automatic speech understanding if prosodic boundary marking is taken into account, because syntactic boundaries are often marked by prosodic means. Because large databases are needed for the training of statistical models for prosodic boundaries, we developed a labeling scheme for syntactic-prosodic boundaries within the German VERMOMIL project (automatic speech-to-speech translation). We compare the results of classifiers (multi-layer perceptrons and language models) trained on these syntactic-prosodic boundary labels with classifiers trained on perceptual-prosodic and purely syntactic labels. Recognition rates of up to 96% were achieved. The bounds that we need to parse consist of 20 words on the average and frequently contain sequences of partial sentence equivalents due to restarts, ellipsis, etc. For this material, the boundary scores computed by our classifiers can successfully be integrated into the syntactic parsing of word graphs; currently, they improve the parse time by 92% and reduce the number of parse trees by 96%. This is achieved by introducing a special Prosodic Syntactic Clause Boundary symbol (PSCB) into our grammar and guiding the search for the best word chain with the prosodic boundary scores.

1. INTRODUCTION

Prosody structures utterances and helps the listeners to understand and disambiguate their meaning. To our knowledge, however, so far nobody has really integrated this information into a complete automatic speech understanding system. We will present a syntactic analysis of word hypotheses graphs using prosodic clause boundary information. Our research is carried out in the speech-to-speech translation project VERMOMIL [19, 6] (domain: appointment scheduling) where the influence of prosody can already be evaluated in an end-to-end system: for the integration of prosody in the VERMOMIL system, cf. [12], for the linguistic processing of VERMOMIL, cf. [4].

A corpus analysis of VERMOMIL data (human-human dialog) showed that about 70% of the utterances contain more than one sentence [15]. About 25% of the utterances are longer than 10 seconds. Especially for such a material, the use of prosody in parsing is crucial for two reasons:

First, to ensure that most of the words that were spoken are recognized, a large word hypotheses graph (currently about 10 hypotheses per spoken word) has to be generated. Finding the correct (or approximately correct) path through a word hypotheses graph is thus an enormous search problem.

Second, spontaneous speech contains many elliptic constructions. So even if the spoken word sequence has been recovered by word recognition correctly, there might be many different parses possible, especially with longer turns. Consider the following two of the at least 96 different syntactic readings for a word sequence taken from the VERMOMIL corpus: “Ja zur Not. Geh’s auch am Samstag?” vs. “Ja zur Not. Geh’s auch am Samstag.” The appropriate English translations are “O.K., if necessary. Is Saturday possible as well?” vs. “Well, if necessary, Saturday is possible as well.”

In these examples, only the prosodically marked boundaries can disambiguate between the two different semantic meanings and pragmatic interpretations.

We use prosody only to guide the search for the best syntactic parse through the word graph; no hard decisions are made. Partial parses are ranked in an agenda according to a score which takes into account the prosodic probability for a clause boundary. At each step of the search the best partial parse is extended. So the main use of prosodic information will be to speed-up the search for the best complete parse. However, in a system with limited resources (i.e. the syntax has to produce a parse after n turns length or it will receive a time out signal), this speed-up will also increase the recognition rate of the syntax module.

2. PROSODIC SYNTACTIC BOUNDARY MARKERS — THE M-LABEL SYSTEM

We developed a syntactic-prosodic labeling scheme for German that provides a coarse labeling of syntactic boundaries. It can be done fast and fairly reliable because it is based solely on the transcriptions of the turn; i.e., we do not have to listen to the turns. Prosodic knowledge is used, i.e., syntactic boundaries are marked differently depending on whether they are likely to be marked prosodically. Typical spontaneous speech phenomena are taken into account as well. Currently we distinguish 10 labels which are grouped into three major classes:

M3: clause boundary (between main clauses, subordinate clauses, elliptic clauses, etc.)

M0: no clause boundary

M0: undefined, i.e. M3 or M0 cannot be assigned to this word boundary without context knowledge and/or perceptual analysis.

The labeling scheme is described in more detail in [2, 3]. In [2] we compared these labels with purely prosodic labels (S-labels) [14], and precise syntactic labels (S-labels) [7].

1In this work was partly funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the VERMOMIL Project under Grant 011 IV 101 AU and funded under Grants 011 IV 102 F/4 and 011 IV 102 H/0. The responsibility for the contents lies with the author.

To appear in Proc. ICASSP-97, Apr 21-24, Munich, Germany © IEEE 1997
This comparison showed that there is a high agreement between these labels and, hence, justifies our rather coarse labeling scheme. The advantage of the M-labels is that a high number of labeled data can be produced within a short time, because they do not require a complete syntactic analysis and they do not rely on perceptual evaluation. Meanwhile, there are 7,286 turns (about 150,000 words) labeled with the Ms, which took only a few months.

3. SPEECH DATABASE

For the classification experiments in Section 4, we used 3 dialogs of the Verbmobil database for testing (64 turns of 3 male and 3 female speakers, 1513 words, 12 minutes in total). For the training of the multi-layer perceptron (MLP) all the available data labeled with the E-labels were used (757 turns) except for the test set, for the language model (LM), trained with the M-labels, 6297 turns were used. For the parsing experiments in Section 5 we chose 594 turns out of 122 dialogs. These turns had been selected for evaluation purposes by the DFKI (Saarbrücken), who was responsible for the integration of the Verbmobil demonstrator. For all of these turns, word graphs were provided by DFKI using the word recognizer of the University of Karlsruhe. The word graphs contained 9.3 hypotheses per spoken word. The word accuracy, i.e., the highest accuracy of any of the paths contained in the graph, was 73.5%. 117 word graphs were correct, i.e., they contained the spoken word chain.

4. AUTOMATIC BOUNDARY CLASSIFICATION

We will now compare classification results obtained with a multi-layer perceptron (MLP), a stochastic (n-gram) language model (LM), and a combination of both classifiers. The MLP serves as an acoustic-prosodic classifier getting acoustic and few lexical features as its input. The LM estimates probabilities for boundaries given a few words in the context of the word. With these classifiers for each of the words in a word chain or in a word graph a probability for a clause boundary being after the word is computed.

The computation of the acoustic-prosodic features is based on an automatic time alignment of the phoneme sequence corresponding to the spoken or recognized words. For the boundary classification experiments we only use the aligned spoken words thus simulating 100% word recognition. For each word a vector of prosodic features is computed automatically from the speech signal. The feature set is described in [9] and, in more detail, in [2]. In order to balance for the priori probabilities of the different classes, during training the MLP was presented with an equal number of feature vectors from each class. For the experiments, MLPs with 40/20 nodes in the first/second hidden layer showed best results. During training B3 vs. ~B3 was taken as reference.

Trigram language models (LM) were additionally used for the classification of boundaries. They model partial word chains where M3 and M0 boundaries have been inserted. This method as well as the combination of LM and MLP scores is described in more detail in [11, 10].

In Table 1, we compare the results of different classifiers for the two main classes boundary vs. not-boundary using two different types of reference boundaries: B, M, and MB, which is a combination of both. In the case of M3 vs. M0, the `undefined' boundaries M0 are not taken into account. As for MB, MB3 represents all word boundaries which are classifier labeled with M3 or with MU and B3, MB5 refers to all other word boundaries. These combined labels represent best what the syntax would like to get delivered by the prosody. The first number in each row of the table shows the overall recognition rate, the second is the average of the class-wise recognition rates. The recognition rates take all word boundaries except the end of turns into account; the latter can be classified in a trivial way. It can be noticed that, roughly, the results get better from top left to bottom right. The best results were obtained when using the MLP with the LM no matter whether the perceptual B or the syntactic-prosodic M labels serve as reference. The LM alone is already very good; we have, however, to consider that it cannot be applied to the `undefined' classes MB which are of course not true for a correct syntactico-semantic processing and which account for 4% of all word boundaries and for 23% of all non-M0 boundaries. Especially for these cases, we need a classifier trained with perceptual-prosodic labels. Note, however, that even on the M2/M0-task the combination of the two classifiers, MLP+LM, shows slightly better results than the LM alone.

Due to the different a priori probabilities, the boundaries are recognized worse than the non-boundaries with the LMs (e.g., 80.8% for M3 vs. 97.7% for M0 for the MLP+LM classifier); this causes the lower average of the class-wise recognition rates compared to the overall recognition rates. It is of course possible to adapt the classification to various demands, e.g., in order to get better recognition rates for the M3 boundaries if more false alarms can be tolerated.

In the following section, prosodically scored word graphs are used for parsing. This means, that for each of the word hypotheses contained in the graph the probability for a clause boundary following this word is computed. The computation of the acoustic features as well as of the LM score is based on 2±2 context words. In the case of the word graphs, the best scored word hypotheses being in the context of a word hypothesis are used. This approach is sub-optimal, but we could show in [11], that the recognition rate does not decrease very much when classifying word graphs instead of the spoken word chain.

5. GRAMMAR AND PARSER

In this paper, we describe the interaction of prosody with the syntax-module developed by Siemens (Munich); for the interaction with another syntax-module developed by IBM (Heidelberg) cf. [1]. In the module described here, we use a Trace and Unification Grammar (TUG) [5] and a modification of the parsing algorithm of Tomita [17]. The basis of a TUG is a context-free grammar augmented with PATR-II-style feature equations. The Tomita parser uses a graph-structured stack as central data structure [16]. After processing word w, the top nodes of this stack keep track of all partial derivations for w,...w. In [15], a parsing scheme for word graphs is proposed using this parser to combine different knowledge sources when searching the word graph for the optimal word sequence: a TUG, a statistical trigram or bigram model and the score of the acoustic component. In the work described here we added another knowledge source for clause boundaries compared as indicated in Section 4.

When searching the word graph, partial sentence hypotheses are organized as a tree. A graph-structured stack of the Tomita parser is associated with each node. In the
Table 2. Grammar 1 for multiple phrase utterances

<table>
<thead>
<tr>
<th>Rule</th>
<th>Input</th>
<th>Phrase</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>(rule 1)</td>
<td>input</td>
<td>phrase</td>
<td>input</td>
</tr>
<tr>
<td>(rule 2)</td>
<td>phrase</td>
<td>s</td>
<td>PSB</td>
</tr>
<tr>
<td>(rule 3)</td>
<td>phrase</td>
<td>s_all</td>
<td>PSB</td>
</tr>
<tr>
<td>(rule 4)</td>
<td>phrase</td>
<td>np</td>
<td>PSB</td>
</tr>
<tr>
<td>(rule 5)</td>
<td>phrase</td>
<td>excl</td>
<td>PSB</td>
</tr>
<tr>
<td>(rule 6)</td>
<td>phrase</td>
<td>excl</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Grammar 2 for multiple phrase utterances

<table>
<thead>
<tr>
<th>Rule</th>
<th>Input</th>
<th>Phrase</th>
<th>PSB</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>(rule 7)</td>
<td>input</td>
<td>phrase</td>
<td>PSB</td>
<td>input</td>
</tr>
<tr>
<td>(rule 8)</td>
<td>phrase</td>
<td>s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(rule 9)</td>
<td>phrase</td>
<td>s_all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(rule 10)</td>
<td>phrase</td>
<td>np</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. EXPERIMENTAL RESULTS

In experiments using a preliminary version of the subgrammars for the individual types of phrases, we compared the grammar explained in Section 5 with a grammar that obligatorily required a PSB behind every input phrase, see Table 3.

With the grammar shown in Table 2 149 word graphs could successfully be analysed: with the one given in Table 3, only 79 word graphs were analysed. This indicates that often the prosodic module computes a high score for PSB after exclamatory particles so that parsing fails if a PSB is obligatorily required as in the grammar of Table 3. With an improved version of the grammar for the individual phrases, we repeated the experiments using the grammar of Table 2 and compared them with the parsing results using a grammar without PSBs. For the latter, we took the category PSB out of the grammar and allowed all input phrases to adjoin recursively to each other. The graphs were parsed without taking notice of the prosodic PSB information contained in the lattice. In this case, the number of readings increases and the efficiency decreases drastically, cf. Table 4. The statistics show that on the average, the number of readings decreases by 96% when prosodic information is used, and the parse time drops by 97%. If the lattice parser does not pay attention to the information on possible PSBs, the grammar has to determine by itself where the phrase boundaries in the utterance

4For this word chain, it would make no difference for the text understanding component, whether the PSB is before or after Dienstag. Actually, the spoken word chain is: Ja, das paßt. Nur Dienstag ist der fünftehnte. and the dialog goes like this: A: What about Tuesday the sixteenth? B: Yes, that's ok. But Tuesday is the fifteenth. A: Sorry. Then let's say Wednesday the sixteenth. B: OK, Fine. B thus only confirms the sixteenth, but not Tuesday.
Table 4. Parsing statistics for 594 word graphs

<table>
<thead>
<tr>
<th></th>
<th>with PSCS</th>
<th>without PSCS</th>
</tr>
</thead>
<tbody>
<tr>
<td># successful analyses</td>
<td>329</td>
<td>368</td>
</tr>
<tr>
<td># syntactic readings</td>
<td>5.6</td>
<td>13.7</td>
</tr>
<tr>
<td>parse time (sec)</td>
<td>3.1</td>
<td>38.6</td>
</tr>
</tbody>
</table>

Table 5. Syntactically possible segmentations

| [er, kommt, morgen] | He comes tomorrow. |
| [er, [kommt, morgen]| He? Comes tomorrow! |
| [er, [kommt], [morgen]| He? Comes! Tomorrow! |

The fact that 9 word graphs (i.e. 2%) could not be analyzed with the use of prosody is due to the fact that the search space is explored differently and that the fixed time limit has been reached before the analysis succeeded. However, this small number of non-analyzable word graphs is neglectable considering the fact that without prosody, the average real-time factor is 6.1 for the parsing. With prosodic information the real-time factor drops to 0.5: the real-time factor for the computation of prosodic information is 1.0 (with word graphs of about 10 hypotheses per spoken word).

Empty categories are an even more serious problem. They are used by the grammar in order to deal with verb movement and topicalization in German. The binding of these empty categories has to be checked inside a single input phrase, i.e., the main sentence. No movement across phrase boundaries is allowed. Now, whenever a PSCS signals the occurrence of a boundary, the parser checks whether all binding conditions are satisfied and accepts or rejects the path that was found so far. This mechanism works efficiently in the case prosodic information was used. For the grammar without PSCSs, no signal where to check the binding restrictions is available. Therefore, the uncertainty about segmentation of multiple phrase utterances led to indefinite parsing time for some of the lattices in the corpus. Those lattices were analyzed correctly with PSCSs.

7. CONCLUSION

We showed that prosodic clause boundary information can reduce the parse time of word graphs computed for spontaneous speech by 96%. The number of parse trees of the resulting analysis decreases by 96%. This is especially due to the high number of elliptic and interrupted phrases contained in spontaneous speech, which cause that the position of clause boundaries is highly ambiguous. Apart from differences in the particular technical solutions of some subproblems, our approach differs from the prosodic parse-rescoring described in [13, 8] mainly in the fact that we first compute prosodic scores based on the word hypotheses generated by the word recognizer. These scores are then integrated directly into the parsing process which does not only reduce the number of readings but also the parse time.

REFERENCES

[2] A. Batliner, R. Kompe, A. Kiefeling, M. Mast, and E. Nöth. All about Ms and Is, not to forget As, and a comparison with Bs and Ss and Ds. Verbmobil Memo 102, 1996.