PROSODY, EMPTY CATEGORIES AND PARSING — A SUCCESS STORY —

A. Batliner[1], *A. Feldhaus*[2], *S. Geißler*[2], *T. Kiss*[2], *R. Kompe*[3], *E. Nöth*[3]

[1]L.-M.-Universität, Institut für Deutsche Philologie, 80799 München, Germany
Anton.Batliner@phonetik.uni-muenchen.d400.de
[2]IBM Informationssysteme GmbH, Institut für Logik und Linguistik, 69020 Heidelberg, Germany
[3]Universität Erlangen-Nürnberg, Lehrstuhl für Mustererkennung, 91058 Erlangen, Germany

ABSTRACT

We describe a number of experiments that demonstrate the usefulness of prosodic information for a processing module which parses spoken utterances with a feature-based grammar employing empty categories. We show that by requiring certain prosodic properties from those positions in the input, where the presence of an empty category has to be hypothesized, a derivation can be accomplished more efficiently. The approach has been implemented in the machine translation project Verbmobil and results in a significant reduction of the work-load for the parser.

1. INTRODUCTION

In this paper we describe how syntactic and prosodic information interact in a translation module for spoken utterances which tries to meet the two – often conflicting – main objectives, the implementation of theoretically sound solutions and efficient processing of the solutions. As an analysis which meets the first criterion but seemingly fails to meet the second one, we take an analysis of the German clause which relies on traces in verbal head positions in the framework of Head-driven Phrase Structure Grammar (Hpsg, cf. [8]).

The methods described in this paper have been implemented as part of the IBM-SynSem-Module and the FAU-Erlangen/LMU-Munich–Prosody-Module in the MT project Verbmobil (cf. [9]) where spontaneously spoken utterances in a negotiation dialogue are translated. In this system, an Hpsg is processed by a bottom-up chart parser that takes word graphs as its input. In a preprocessing step it is searched for alternative string hypotheses contained in the graph. They differ in the wording and in the positions of empty elements and of segment boundaries. The individual segments are then parsed one after the other. The output of the parser is the semantic representation for the best string hypothesis in the graph. It is our main result that prosodic information can be employed in such a system to determine possible locations for empty elements in the input. Rather than treating prosodic information as virtual input items which have to match an appropriate category in the grammar rules [3], or which by virtue of being 'unknown' in the grammar force the parser to close off the current phrase [6], our parser employs prosodic information as affecting the postulation of empty elements. An extended description of the HPSG analysis and the processing of empty elements can also be found in [1]. In the present paper new results concerning the parsing of graphs are presented, where prosodic information is also used for the segmentation of string hypotheses. Furthermore, for the first time we employ the combination of a trigram language model with an acoustic–prosodic classifier in the syntactic analysis.

2. AN HPSG ANALYSIS OF GERMAN CLAUSE STRUCTURE

Hpsg makes crucial use of "head traces" to analyze the verb-second (V2) phenomenon pertinent in German, i.e. the fact that finite verbs appear in second position in main clauses but in final position in subordinate clauses, as exemplified in (1a) and (1b).

1(a) Gestern reparierte er den Wagen.
(Yesterday fixed he the car)
'Yesterday, he fixed the car.'
1(b) Ich dachte, daß er gestern den Wagen reparierte.
(I thought that he yesterday the car fixed)
'I thought that he fixed the car yesterday'.

Following [5] we assume that the structural relationship between the verb and its arguments and modifiers is not affected by the position of the verb. The overt relationship between the verb '*reparierte*' and its object '*den Wagen*' in (1b) is preserved in (1a), although the verb shows up in a different position. The apparent contradiction is resolved by assuming an empty element which serves as a substitute for the verb in second position. The empty element fills the position occupied by the finite verb in subordinate clauses, leading to the structure of main clauses exemplified in Fig. 1.

[1] This work was partly funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant 01 IV 101 V and funded under Grants 01 IV 102 F/4 and 01 IV 102 H/0. The responsibility for the contents lies with the authors. We would like to thank Thomas Kemp, ILKD, Universität Karlsruhe for providing us with the word graphs used in the experiments described in this paper.

4th International Conference on Spoken
Language Processing (ICSLP 96)
Philadelphia, PA, USA
October 3-6, 1996

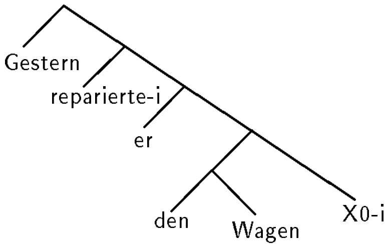

Figure 1: Syntax tree for 'Gestern reparierte er den Wagen.'

The empty verbal head in Fig. 1 carries syntactic and semantic information. Particularly, the empty head licenses the realization of the syntactic arguments of the verb according to the rule schemata of German and HPSG's Subcategorization Principle. The structure of the main clause presented in Fig. 1 is widely assumed in the linguistic literature and can be justified on syntactic and semantic grounds – for a detailed discussion see [5]. Technically this analysis is implemented by ensuring that each lexical entry for a finite verb in second (or first) position in a sentence is associated with a corresponding empty head. A special feature – the DSL (Double Slash) feature – establishes the necessary connection between these two structures and ensures the percolation of grammatical information indicated by the coindexation of *reparierte* and *X0* in Fig. 1.

3. PROCESSING EMPTY ELEMENTS

Direct parsing of empty elements can become a tedious task, decreasing the efficiency of a system considerably. Note first, that a reduction of empty elements in a grammar in favor of disjunctive lexical representations, as suggested in [8], cannot be pursued. [8] assume that an argument may occur on the SUBCAT or on the SLASH list. A lexical operation removes the argument from SUBCAT and puts it onto SLASH. Hence, no further need for a syntactic representation of empty elements emerges. This strategy, however, will not work for head traces because they do not occur as dependents on a SUBCAT list. If empty elements have to be represented syntactically, a top-down parsing strategy seems better suited than a bottom-up strategy. Particularly, a parser driven by a bottom-up strategy has to hypothesize the presence of empty elements at every point in the input. In HPSG, however, only very few constraints are available for a top-down regime since most information is contained in lexical items. The parser will not restrict the stipulation of empty elements until a lexical element containing restrictive information has been processed. The apparent advantage of top-down parsing is thus lost when HPSGs are to be parsed. The same criticism applies to other parsing strategies with a strong top-down orientation, such as left corner parsing or head corner parsing.

We have thus chosen a bottom-up parsing strategy where the introduction of empty verbal heads is constrained by syntactic and prosodic information. The syntactic constraints build on the facts that a) a verb trace will occur always to the right of its licenser and b) always 'lower' in the syntax tree. Furthermore c) since the DSL percolation mechanism ensures structure sharing between the verb and its trace, a verb trace always comes with a corresponding overt verb. Although a large number of bottom-up hypotheses regarding the position of an empty element can be eliminated by providing the parser with the aforementioned information, the number of wrong hypotheses is still significant. In a verb-2nd clause most of the input follows a finite verb form so that condition a) indeed is not very restrictive. Condition b) rules out a large number of structures but often cannot prevent the stipulation of traces in illicit positions. Condition c) has the most restrictive effect in that the syntactic potential of the trace is determined by that of the corresponding verb. If the number of possible trace locations could be reduced significantly, the parser could avoid a large number of subanalyses that conditions a)-c) would rule out only at later stages of the derivation. The strategy that will be advocated in the remainder of this paper employs prosodic information to accomplish this reduction.

Empty verbal heads can only occur in the right periphery of a phrase, i.e. at a phrase boundary. The introduction of empty arcs is then not only conditioned by the syntactic constraints mentioned before, but additionally, by certain requirements on the prosodic structure of the input. It turns out, then, that a fine-grained prosodic classification of utterance turns, based on correlations between syntactic and prosodic structure is not only of use to determine the segmentation of a turn, but also, to predict which positions are eligible for trace stipulation. The following section sketches the prosodic classification, section 5 features the results of the current experiments.

4. PROSODIC BOUNDARY CLASSIFICATION

For the segmentation of turns into syntactically meaningful units, we used two classifiers: Mulit–layer perceptrons (MLP) were trained based on perceptual–prosodic boundaries using a large prosodic feature vector. Trigram language models (LM) were trained with word chains annotated with coarse syntactic boundaries. Both methods are outlined in [2]. With this the probability for a boundary being after each of the words in a turn was computed. The test set comprises 3 dialogs (64 turns of 3 male and 3 female speakers, 12 minutes in total). Each word boundary was manually labeled with S3+ (syntactic main boundary), S3– (no syntactic main boundary), S3? (ambiguous boundary). The LM alone yielded already good recognition results because it was trained with a very large data base; the combination with an MLP improved the recognition rate further, and it is especially needed for the classification of the S3? boundaries that cannot be covered by the LM. We obtained an overall recognition rate of 94% (average of the class–wise recognition rates: 86%) for the two classes S3+ and S3– not counting the turn final boundaries.[1].

5. RESULTS

All experiments were conducted on word graphs with more than 5 (experiments 1–3) and 10 hypotheses (experiment 4) per spoken word. In the first two experiments the word graph was parsed without the use of prosodic information and then the word chain found during the parse was manually evaluated. The third and fourth experiments compare parse times of word graphs. The word graphs used in the fourth experiment were obtained from real spontaneous dialogs. All other word graphs were generated during tests of non–naive persons with the VERBMOBIL demonstrator. The hypotheses in the word graphs were automatically annotated with probabilities for S3+ using the classifier described in Section 4.

Experiment 1: In order to approximate the usefulness of prosodic information to reduce the number of verb trace hypotheses for the parser we examined a corpus of 104 utterances with prosodic annotations denoting the probability of a syntactic boundary after every given word. For every word hypothesis where the S3+ boundary probability exceeds an experimentally optimized threshold value, we considered the hypothesis that this node is followed by a verb trace. These hypotheses were then rated valid or invalid by the grammar writer. The observations were rated according to Table 3[2]: Evaluation of these figures for our test corpus yielded the results presented in Table 4. In practice this means that the number of locations where the parser has to assume the presence of a verb trace could be reduced by 63% from 1121 to 412 while only 6 (4%) necessary trace positions remained unmarked. These results were obtained from a corpus of spoken utterances many of which contained several independent phrases and sentences. These, however, are also often separated by an S3+ boundary, so that the error rate is likely to drop considerably if a segmentation of utterances into syntactically well-formed phrases is performed prior to the trace detection (cf. experiment 2). Since cases where the verb trace is not located at the end of a sentence (i.e. where extraposition takes place) involve a highly characteristic categorial context, we achieved a further improvement by the trace/no-trace classification based on prosodic information combined with a language model (cf. Table 7 below).

The problem with the approach described so far is that a careful estimation of the threshold value is necessary and this threshold may vary from speaker to speaker or between certain discourse situations. The analysis fails in those cases where the correct position is rated lower than this value, i.e. where the parser does not consider the correct trace position at all. Thus, in a second experiment we examined how the syntactically correct verb trace position is ranked among the positions proposed by the prosody module w.r.t. its S3+ boundary probability. If the correct position turns out to be consistently ranked among the positions with the highest S3+ probability within a sentence then it might be preferable for the parsing module to consider the S3+ positions in descending order rather than to introduce traces for all positions ranked above a threshold.

	X0 position	no X0 position
X0 prop.	Correct: 138	False Alarm : 274
no X0 prop.	Miss : 6	X : 703

Table 3: Classification results for verb trace positions

Recall	=	$\frac{Correct}{(Correct+Miss)}$	=	95.8
Precision	=	$\frac{Correct}{(Correct+False)}$	=	33.5
Error	=	$\frac{(Miss+False)}{(Correct+False+Miss+X)}$	=	25.0

Table 4: Results for the identification of possible verb trace positions.

Experiment 2: We considered only those segments in the input that represent V2 clauses, i.e. we assumed that the input has been segmented correctly. Within these 134 sentences we ranked all the spaces between words according to the associated S3+ probability and determined the rank of the correct verb trace position. Table 5 shows that in the majority of cases the position with the highest S3+ probability turns out to be the correct one. It has to be added though, that in many cases the correct verb trace position is at the end of the sentence which is often very reliably marked with a prosodic phrase boundary, even if this sentence is uttered in a sequence together with other phrases or sentences. This end-of-sentence marker will be assigned a higher S3+ probability in most cases, even if the correct verb trace position is located elsewhere.

Rank	1	2	3	4	5	6	7	≥ 7
# of occ.	96	22	7	4	3	0	1	1

Table 5: Ranking of the syntactically correct verb trace position within a sentence according to the S3+ probability.

Experiment 3: Now, we were interested in the overall speedup of the processing module that resulted form our approach. In order to estimate this, we parsed a corpus of 109 turns in two different settings: While in the first round the threshold value was set as described above, we selected a value of 0 for the second pass. The parser thus had to consider every position in the input as a potential head trace location just as if no prosodic information about syntactic boundaries were available at all. Employing prosodic information reduces the parser runtime for the corpus by about 46%, cf. Table 6.

[1] Note that this combination of MLP and LM was only used in the fourth experiment. Furthermore, for the first three experiments reported in the following, an older version of the MLP trained on a smaller data base was used. It achieved an average recognition rate of 85%. Therefore, if these experiments were repeated even better results could be expected.

[2] *X0 position* means that the relevant position is occupied by a X0 gap, *X0 prop.* means that the classifier proposes an X0 at this position.

	With Prosody	Without Prosody
Overall	704.8	1304.2
Average	6.5	11.9
Speedup	45.96%	./.

Table 6: Runtimes (in secs) for parsing batch-jobs with and without the use of prosodic information

Experiment 4: Finally, we were interested in the effect of employing a language model (LM) in combination with the MLP classifier. In this setting not only the prosodic/acoustic properties of the input but also the surrounding word forms play a role in determining the probability of the presence or absence of a phrase boundary. Such an LM can be expected to exhibit a significant effect on the distribution of these probabilities: The position immediately following a determiner, e.g., will be marked with a low probability for a phrase boundary. For this experiment, a sample corpus with word graphs for 21 turns containing about 10 word hypotheses per spoken word was automatically annotated with probabilities for phrase boundaries with and without using the LM information as described above. As expected, without the LM information the number of positions that received a high S3+ probability was much larger than in the setting which did make use of the LM, because the LM accounts also for the a priori probabilities. However, as mentioned above, the boundary probabilities not only interact with the hypothesized presence or absence of empty elements, but also play a key role in the segmentation of turns into syntactically autonomous units (segments). As can be seen in Table 7, not using the LM information leads the processing module to hypothesize a larger number of segment boundaries that are less meaningful in the context of a further processing in the full system than those that can be identified in the alternative setting. The seemingly unfavorable result that the parser performed much slower on the LM corpus has to be interpreted in the light of the observation that in the non-LM setting the input was segmented into about twice as many units. On the average the units are twice as large in the with-LM setting, so that it can be seen as a positive result that the overall runtime increased only by a factor of two. Furthermore, for a subsequent analysis of the units in the full VERBMOBIL system the larger units are much more useful as a manual evaluation of the parsing results showed. This is also indicated by the fact that in the no-LM setting the average unit length is less than 3 words. With the LM even a larger total portion of the input (52% of the units) could be parsed than without the LM (44%). Note, that without prosodic information we were not able to parse the corpus at all due to memory limitations.

	MLP+LM	MLP
# of units in turns	60	117
# of successfully parsed units	31	52
# average runtime per unit	17.1	5.1
# overall runtime	1025.8	595.9

Table 7: Batchmode results on corpora with and without an LM boundary classification[3]

6. CONCLUSION

In [7] prosodic information was used to score alternative parses of the same word sequence. Our approach differs from that one, because we use the prosodic information in a preprocessing step where alternative string hypotheses are selected based on prosodic information and are parsed afterwards. We showed that prosodic information can be employed in a speech processing system to determine possible locations of empty elements and of segment boundaries. Although the primary goal of the categorial labelling of prosodic phrase boundaries was to adjust the division of turns into sentences to the intuitions behind the grammar used, it turned out that the same classification can be used to minimize the number of wrong hypotheses pertaining to empty productions in the grammar. We found a very useful correspondence between an observable physical phenomenon – the prosodic information associated with an utterance – and a theoretical construct of formal linguistics – the location of empty elements in the respective derivation. The method has been successfully implemented and tested on real spontaneous speech data.

7. REFERENCES

1. Batliner, A., Feldhaus, A., Geißler, S., Kießling, A., Kiss, T., Kompe, R., Nöth, E. "Integrating Syntactic and Prosodic Information for the Efficient Detection of Empty Categories", *Proc. of the Int. Conf. on Computational Linguistics.*
2. Batliner, A., Kießling, A., Kompe, R., Niemann, H., and Nöth, E. "Syntactic-prosodic Labeling of Large Spontaneous Speech Data-bases", in these proceedings.
3. Bear, J. and Price, P.: "Prosody, Syntax, and Parsing", *Proceedings of the 28th Conf. of the ACL. 1990. pp. 17–22.*
4. Borsley, R. "Phrase Structure Grammar and the Barrier Conception of Clause Structure", *Linguistics, 27. 1989. pp. 843-863.*
5. Kiss, T. and Wesche, B. "Verb order and Head-Movement in German." In Herzog, O., Rollinger, C.-R. (eds.): *Text Understanding in LILOG. Integrating Artificial Intelligence and Computational Linguistics.* Springer, pp. 216-240, Berlin. 1991.
6. Marcus, M. and Hindle, D. "Description Theory and Intonation Boundaries". In: Altmann, G. (ed.), *Cognitive Models of Speech Processing.* MIT Press, Cambridge. 1990. pp. 483–512.
7. Ostendorf, M., Wightman, C.W., Veilleux, N.M. "Parse Scoring with Prosodic Information: an Analysis/Synthesis approach". *Computer Speech and Language*, Vol. 7, No. 3, 1993, pp. 193–210.
8. Pollard, C. and Sag, I.A., *Head-driven Phrase Structure Grammar*, Univ. of Chicago Press, Chicago. 1994.
9. Wahlster, W. "Verbmobil: Übersetzung von Verhandlungsdialogen", VERBMOBIL-*Report 1.* DFKI Saarbrücken. 1993.

[3] Since the data were realistic utterances of spontaneous speech that were automatically annotated with prosodic information and then parsed without any further editing, the rate of about 50% of successes comes out as a reasonable result.