An Experiment On Incremental Analysis Using Robust Parsing Techniques

Kilian Foth and **Wolfgang Menzel** and **Horia F. Pop** and **Ingo Schröder**
foth | menzel | hfpop | schroeder@nats.informatik.uni-hamburg.de
Fachbereich Informatik, Universität Hamburg
Vogt-Kölln-Straße 30, 22527 Hamburg, Germany

Abstract

The results of an experiment are presented in which an approach for robust parsing has been applied incrementally. They confirm that due to the robust nature of the underlying technology an arbitrary prefix of a sentence can be analysed into an intermediate structural description which is able to direct the further analysis with a high degree of reliability. Most notably, this result can be achieved without adapting the grammar or the parsing algorithms to the case of incremental processing. The resulting incremental parsing procedure is significantly faster if compared to a non-incremental best-first search. Additionally it turns out that longer sentences benefit most from this acceleration.

1 Introduction

Natural language utterances usually unfold over time, i. e., both listening and reading are carried out in an incremental left-to-right manner. Modeling a similar type of behaviour in computer-based solutions is a challenging and particularly interesting task for a number of quite different reasons that are most relevant in the context of spoken language systems:

- Without any external signals about the end of an utterance, incremental analysis is the only means to segment the incoming stream of speech input.
- An incremental analysis mode provides for a more natural (mixed-initiative) dialogue behaviour because partial results are already available well before the end of an utterance.
- Parsing may already take place in concurrency to sentence production. Therefore the speaking time becomes available as computing time.
- Dynamic expectations about the upcoming parts of the utterance might be derived right in time to provide guiding hints for other processing components, e. g., predictions about likely word forms for a speech recognizer (Hauenstein and Weber, 1994).

In principle, two alternative strategies can be pursued when designing an incremental parsing procedure:

1. To keep open all necessary structural hypotheses required to accomodate every possible continuation of an utterance. This is the strategy usually adopted in an incremental chart parser (Wirén, 1992).
2. To commit to one or a limited number of interpretations where
   (a) either this commitment is made rather early and a mechanism for partial reanalysis is provided (Lombardo, 1992) or
   (b) it is delayed until sufficient information is eventually available to take an ultimate decision (Marcus, 1987).

The apparent efficiency of human language understanding is usually attributed to an early commitment strategy.

Our approach, in fact, represents an attempt to combine these two strategies: On the one hand, it keeps many of the available building blocks for the initial part of an utterance and passes an *(updated) search space* to the following processing step. Additionally, the *optimal* structural description for the data already available is determined. This not only makes an actual interpretation (together with expectations for possible continuations) available to subsequent processing components, but also opens up the possibility to use this information to effectively constrain the set of new structural hypotheses.

Determining the optimal interpretation for a yet incomplete sentence, however, requires a parsing approach robust enough to analyse an arbitrary sentence prefix into a meaningful structure. Therefore two closely related questions need to be raised:

1. Can the necessary degree of robustness be achieved?
2. Is the information contained in the currently optimal structure useful to guide the subsequent analysis?

To answer these questions and to estimate the potential for search space reductions against a possible loss of accuracy, a series of experiments has been conducted. Sections 2 and 3 introduce and motivate the framework of the experiments. Section 4 describes a number of heuristics for re-use of previous solutions and Section 5 presents the results.

2 Robust Parsing in a Dependency Framework

Our grammar models utterances as *dependency trees*, which consist of pairs of words so that one depends directly on the other. This subordination relation can be qualified by a label (e. g. to distinguish complements from modifiers). Since each word can only depend on one other word, a labeled tree is formed, usually with the finite verb as its root.

The decision on which structure to postulate for an utterance is guided by explicit *constraints*, which are represented as universally quantified logical formulas about features of word forms and partial trees. For instance, one constraint might postulate that a relation labeled as 'Subject' can only occur between a noun and a finite verb to its right, or that two different dependencies of the same verb may not both be labeled 'Subject'. For efficiency reasons, these formulas may constrain individual dependency edges or pairs of edges only. The application of constraints can begin as soon as the first word of an utterance is read; no global information about the utterance is required for analysis of its beginning.

Since natural language input will often exhibit irregularities such as restarts, repairs, hesitations and other grammatical errors, individual errors should not make further analysis impossible. Instead, a *robust* parser should continue to build a structure for the utterance. Ideally, this structure should be close to that of a similar, but grammatical utterance.

This goal is attained by annotating the constraints that constitute the grammar with *scores* ranging from 0 to 1. A structure that violates one or more constraints is annotated with the product of the corresponding scores, and the structure with the highest combined score is defined as the solution of the parsing problem. In general, the higher the scores of the constraints, the more irregular constructions can be analysed. Parsing an utterance with annotated or *soft* constraints thus amounts to multi-dimensional optimization. Both complete and heuristic search methods can be employed to solve such a problem.

Our robust approach also provides an easy way to implement *partial parsing*. If necessary, e. g., an isolated noun labeled as 'Subject' may form the root of a dependency tree, although this would violate the first constraint mentioned above. If a finite verb is available, however, subordinating the noun under the verb will avoid the error and thus produce a better structure. This capability is crucial for the analysis of incomplete utterances.

Different *levels* of analysis can be defined to model syntactic as well as semantic structures. A dependency tree is constructed for each of these levels. Since constraints can relate the edges in parallel dependency trees to each other, having several trees contributes to the robustness of the approach. Altogether, the grammar used in the experiments described comprises 12 levels of analysis and 490 constraints (Schröder et al., 2000).

3 Prefix Parsing with Weighted Constraints

In general, dependency analysis is well-suited for incremental analysis. Since subordinations always concern two words rather than full constituents, each word can be integrated into the analysis as soon as it is read, although not necessarily in the optimal way. Also, the pre-computed dependency links can easily be re-used in subsequent iterations. Therefore, dependency grammar allows a fine-grained incremental analysis (Lombardo, 1992).

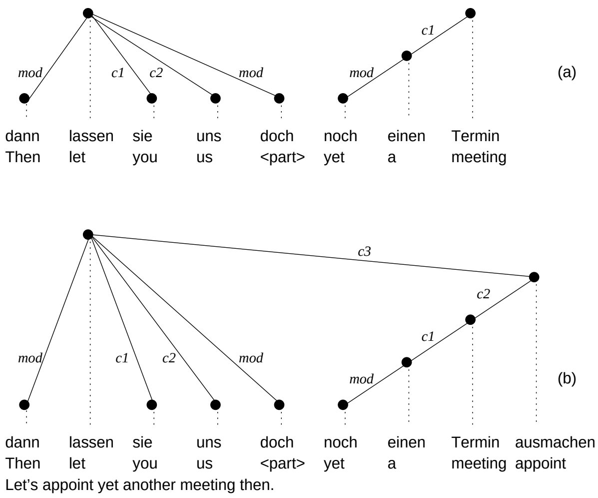

Figure 1: An example for a prefix analysis

When assigning a dependency structure to incomplete utterances, the problem arises how to analyse words whose governors or complements still lie beyond the time horizon. Two distinct alternatives are possible:

1. The parser can establish a dependency between the word and a special node representing a putative word that is assumed to follow in the remaining input. This explicitly models the expectations that would be raised by the prefix. However, unifying new words with these underspecified nodes is difficult, particularly when

multiple words have been conjectured. Also, many constraints cannot be meaningfully applied to words with unknown features.

2. An incomplete prefix can be analyzed directly if a grammar is robust enough to allow partial parsing as discussed in the previous section: If the constraint that forbids multiple trees receives a severe but non-zero penalty, missing governors or complements are acceptable as long as no better structure is possible.

Experiments in prefix parsing using a dependency grammar of German have shown that even complex utterances with nested subclauses can be analysed in the second way. Figure 1a provides an example of this: Because the infinitive verb 'ausmachen' is not yet visible, its complement 'Termin' is analysed as an isolated subtree, and the main verb 'lassen' is lacking a complement. After the missing verb has been read, two additional dependency edges suffice to build the correct structure from the partial parse.

This method allows direct comparison between incremental and non-incremental parser runs, since both methods use the same grammar. Therefore, we will follow up on the second alternative only and construct extended structures guided by the structures of prefixes, without explicitly modeling missing words.

4 Re-Use of Partial Results

While a prefix analysis can produce partial parses and diagnoses, so far this information has not been used in subsequent iterations. In fact, after a new word has been read, another search is conducted on all words already available. To reduce this duplication of work, we wish to narrow down the problem space for these words. Therefore, at each iteration, the set of hypotheses has to be updated:

- By deciding which *old* dependency hypotheses should be *kept*.
- By deciding which *new* dependency hypotheses should be *added* to the search space in order to accomodate the incoming word.

For that purpose, several heuristics have been devised, based on the following principles:

Prediction strength. Restrict the search space as much as possible, while maintaining correctness.

Economy. Keep as much of the previous structure as possible.

Rightmost attachment. Attach the incoming word to the most recent words.

The heuristics are presented here in increasing order of the size of the problem space they produce:

A. Keep all dependency edges from the previous optimal solution. Add all dependency edges where the incoming word modifies, or is modified by, another word.

B. As A, but also keep all links that differ from the previous optimal solution only in their lexical readings.

C. As B, but also keep all links that differ from the previous optimal solution only in the subordination of its last word.

D. As C, but also keep all links that differ from the previous optimal solution only in the subordination of all the words lying on the path from the last word to the root of the solution tree.

E. As D, but for all trees in the previous solution.

5 Results

In order to evaluate the potential of the heuristics described above, we have conducted a series of experiments using a grammar that was designed for non-incremental, robust parsing. We tested the incremental against a non-incremental parser using 222 utterances taken from the VERBMOBIL domain (Wahlster, 1993).

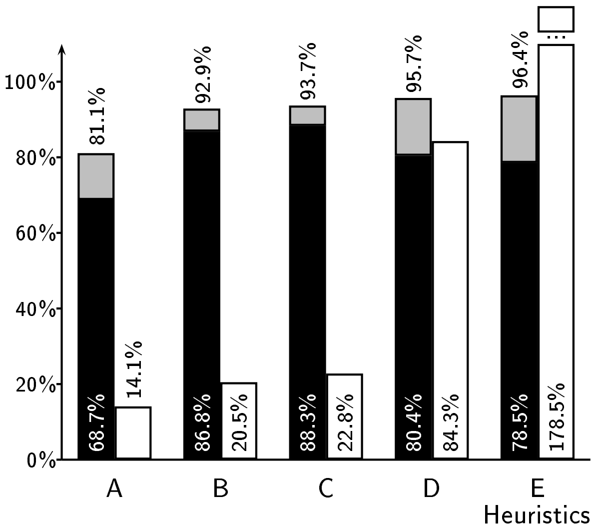

Figure 2: Solution quality and processing time for different heuristics

Figure 2 compares the five heuristics with respect to the following criteria:[1]

Accuracy. The accuracy (gray bar) describes how many edges of the solutions are correct.

$$\text{accuracy} = \frac{\#\ \text{correct edges}}{\#\ \text{edges found}}$$

[1]Note that the heuristics provide at most one solution and may fail to find any solution.

Weak recall. We base our recall measure – given as the black bar – on the number of solutions found non-incrementally (which is less than 100%) because we want focus on the impact of our heuristics, not the coverage of the grammar.

$$\text{weak recall} = \frac{\#\text{ correct edges}}{\#\text{ edges found non-incrementally}}$$

Relative run-time. The run-time required by the incremental procedure as a percentage of the time required by the non-incremental search algorithm is given as the white bar.

The difference between the gray and the black bar is due to errors of the heuristic method, i. e., either because of its incapability to find the correct subordination or due to excessive resource demands (which lead to process abortion).

Two observations can be made: First, all but the last heuristics need less time than the non-incremental algorithm to complete while maintaining a relative high degree of quality. Second, the more elaborate the heuristics are, the longer they need to run (as expected) and the better are the results for the accuracy measure. However, the heuristics D and E could not complete to parse all sentences because in some cases a pre-defined time limit was exceeded; this leads to the observed decrease in weak recall when compared to heuristics C. As expected, a trade-off between computing time and quality can be found. Overall, heuristics C seems to be a good choice because it achieves an accuracy of up to 93.7% in only one fifth of the run-time.

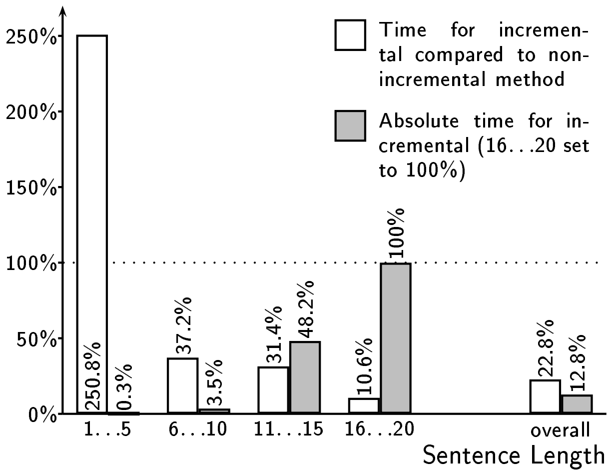

Figure 3: Processing time vs. sentence length

Figure 3 compares the time requirements of heuristics C for different sentence lengths.

- The relative run-time (as in Figure 2) is given as the white bar.
- The gray bar presents the normalized time with the time for sentence length between 16 and 20 set to 100%.

The results show that the speedup observed in Figure 2 is not evenly distributed. While the incremental analysis of the short sentences takes longer (2.5 times *slower*) than the non-incremental algorithm, the opposite is true for longer sentences (10 times *faster*). However, this is welcome behavior: The incremental procedure takes longer only in those cases that are solved very fast anyway; the problematic cases are parsed more quickly. This behavior is a first hint that the incremental analysis with re-use of partial results is a step that alleviates the combinatorial explosion of resource demands.

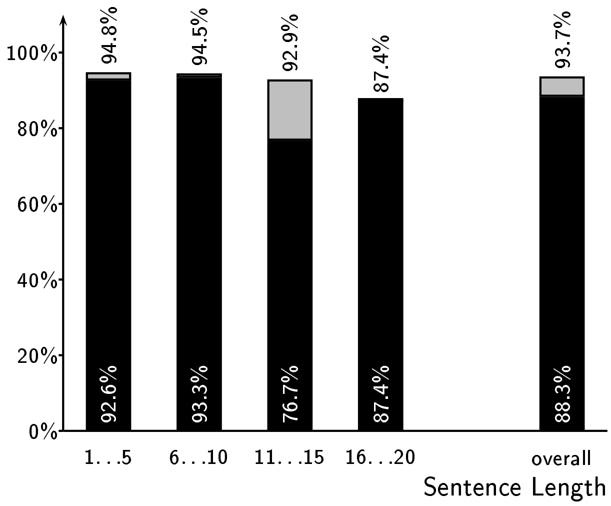

Figure 4: Accuracy vs. sentence length (colors have the same meaning as in Figure 2)

Finally, Figure 4 compares the quality resulting from heuristics C for different sentence lengths. It turns out that, although a slight decrease is observable, the accuracy is relatively independent of sentence length.

6 Conclusions

An approach to the incremental parsing of natural language utterances has been presented, which is based on the idea to use robust parsing techniques to deal with incomplete sentences. It determines a structural description for arbitrary sentence prefixes by searching for the optimal combination of local hypotheses. This search is conducted in a problem space which is repeatedly narrowed down according to the optimal solution found in the preceding step of analysis.

The results available so far confirm the initial expectation that the grammar used is robust enough to reliably carry out such a prefix analysis, although it has originally been developed for the non-incremental case. The optimal structure as determined by the parser obviously contains relevant information about the sentence prefix, so that even very simple and cheap heuristics can achieve a considerable level of accuracy. Therefore, large parts of the search space can be excluded from repeated reanalysis, which eventually makes it even faster than its non-incremental counterpart. Most importantly, the observed speedup grows with the length of the utterance.

On the other hand, none of the used structure-based heuristics produces a significant improvement of quality even if a large amount of computational resources is spent. Quite a number of cases can be identified where even the most expensive of our heuristics is not strong enough, e. g., the German sentence with a topicalized direct object:

Die$_{\text{NOM,ACC}}$	Frau	sieht	der$_{\text{NOM}}$	Mann.
The	woman	sees	the	man.

The woman, the man sees.

Here, when analysing the subsentence *die Frau sieht*, the parser will wrongly consider *die Frau* as the subject, because it appears to have the right case and there is a clear preference to do so. Later, when the next word comes in, there is no way to allow for *die Frau* to change its structural interpretation, because this is not licensed by any of the given heuristics.

Therefore, substantially more problem-oriented heuristics are required, which should take into account not only the optimal structure, but also the conflicts caused by it. Using a weak but cheap heuristics, a fast approximation of the optimal structure can be obtained within a very restricted search space, and then refined by subsequent structural transformations (Foth et al., 2000). To a certain degree this resembles the idea of applying reason maintenance techniques for conflict resolution in incremental parsing (Wirén, 1990). In deciding which strategy is good enough to find the necessary first approximation the results of this paper might play a crucial role, since the possible contribution of individual heuristics in such an extended framework can be precisely estimated.

Acknowledgements

This research has been partly funded by the German Research Foundation "Deutsche Forschungsgemeinschaft" under grant no. Me 1472/1-2.

References

Kilian Foth, Wolfgang Menzel, and Ingo Schröder. 2000. A transformation-based parsing technique with anytime property. In *Proceedings of the International Workshop on Parsing Technologies (IWPT-2000)*, pages 89–100, Trento, Italy.

Andreas Hauenstein and Hans Weber. 1994. An investigation of tightly coupled time synchronous speech language interfaces using a unification grammar. In *Proceedings of the 12th National Conference on Artificial Intelligence: Workshop on the Integration of Natural Language and Speech Processing*, pages 42–49, Seattle, Washington.

Johannes Heinecke, Jürgen Kunze, Wolfgang Menzel, and Ingo Schröder. 1998. Eliminative parsing with graded constraints. In *Proceedings of the Joint Conference COLING/ACL-98*, Montréal, Canada.

Vincenzo Lombardo. 1992. Incremental dependency parsing. In *Proceedings of the Annual Meeting of the ACL*, Delaware, Newark, USA.

Mitchell P. Marcus. 1987. Deterministic parsing and description theory. In Peter Whitelock, Mary McGee Wood, Harold Somers, Rod Johnson, and Paul Bennett, editors, *Linguistic Theory and Computer Applications*, pages 69–112. Academic Press, London, England.

Wolfgang Menzel and Ingo Schröder. 1998. Decision procedures for dependency parsing using graded constraints. In Sylvain Kahane and Alain Polguère, editors, *Proceedings of the Joint Conference COLING/ACL-98 Workshop: Processing of Dependency-based Grammars*, pages 78–87, Montréal, Canada.

Ingo Schröder, Wolfgang Menzel, Kilian Foth, and Michael Schulz. 2000. Modeling dependency grammar with restricted constraints. *International Journal Traitement Automatique des Langues: Grammaires de dépendance*, 41(1).

Wolfgang Wahlster. 1993. Verbmobil: Translation of face-to-face dialogs. In *Proceedings of the 3rd European Conference on Speech Communication and Technology*, pages 29–38, Berlin, Germany.

Hans Weber. 1995. LR-inkrementelles probabilistisches Chartparsing von Worthypothesengraphen mit Unifikationsgrammatiken: Eine enge Kopplung von Suche und Analyse. Verbmobil-Report 52, Universität Erlangen-Nürnberg.

Mats Wirén. 1992. *Studies in Incremental Natural-Language Analysis*. Ph.D. thesis, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

Mats Wirén. 1990. Incremental parsing and reason maintenance. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, pages 287–292, Helsinki, Finland.