

Grammatical Heads Optimized for Parsing and Their Comparison with Linguistic Intuition

Vladimír Kadlec and Pavel Smrž

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
Email: xkadlec@fi.muni.cz, smrz@fi.muni.cz

Abstract. This paper deals with head-driven chart parsing on large natural-language grammars. We present a procedure that optimizes positions of heads in the grammar rules based on the number of edges in the resulting chart. New performance evaluation tool PACE is briefly introduced first. The head-optimizing procedure is described and the results are given. The last part compares the head positions obtained automatically with those set according to traditional linguistics.

1 Introduction

The popularity of head-driven approaches to parsing has grown in the last decade [1]. They have been successfully applied to various kinds of grammars as they can avoid proliferation of partial rule instantiations (i. e. active edges in a chart parser) with rules that contain very unspecific argument positions [2]. As in other “island-oriented” approaches in parsing, the direction of the parsing process is not unidirectional (e. g. from left to right). It starts at a selected symbol on the right-hand side of the given grammar rule – the head – and processes it bidirectionally to the first and to the last rule symbol.

It is usually expected that the position of heads is set according to the linguistic intuition; that the parser should instantiate first the linguistic heads, or governing nodes [3,4]. However, only a limited number of experiments have been published till now that would prove validity of such a heuristics. One of the reasons for this state is that the implementations of head-driven parsing algorithms efficient enough to enable experiments of this kind have appeared only recently.

This paper reports experiments aiming at choosing the best position of rule heads for parsing with various natural-language grammars. The optimization step focuses on the number of edges in the resulting chart. PACE – a newly designed and implemented parser-evaluation tool provides the platform for our resource-demanding experiments.

The optimization can have almost a negligible effect when parsing with small or medium-size grammars. However, as the results that are presented in the paper clearly demonstrate, it can have a crucial importance for parsing with large and highly ambiguous grammars. The Czech grammar used in the described experiments contains approximately 3,000 rules and, if only the feature agreement in noun groups is expanded to the CF backbone [5], it can have as much as 30,000 CF grammar rules. The enormous number of rules is also typical for the “treebank grammars” automatically extracted from syntactically annotated corpora and used for training stochastic parsers (e. g. the PT grammar generated from the Penn Treebank [6]).

The goal to optimize positions of rule heads for parsing is not completely new. It appeared as a rule instantiation strategy in [2]. Oepen and Callmeier discuss head selection procedure in the context of HPSG parsing and employ their PET platform for the evaluation of the results. The form of the LinGO grammar used in the experiments is rather simple, it contains only binary rules (up to two symbols on the right-hand side of rules). On the contrary, at least the Czech grammar used in our experiments contains many rules with complex right-hand sides. It makes the optimizing procedure much more demanding. However, the general conclusion is the same in both cases.

In the experiments that are behind the effort discussed in this paper, a special attention is paid to parser robustness. If no complete parse is found for an input (e.g. from a speech recognizer in a dialogue system) a special technique is employed to efficiently retrieve a set of the most probable maximal subtrees (chunks) to provide a partial analysis of the input. Therefore, we are not able to apply the most popular (and, in general case, efficient) approach to head-driven parsing – head-corner chart parser [3,7] that would prune chart edges that could be needed in our later processing of incomplete parse. Moreover, the head-driven bottom-up algorithm discussed in the paper is also more suitable for our research on incremental parsing. Nevertheless, the head-selection procedure presented in our paper is directly applicable to the head-corner case.

The rest of the paper is organized as follows. The next section discusses the architecture of PACE. Then, we present the optimizing procedure that aims at reduction of edges in the resulting chart. The Czech and ATIS testing grammars are briefly introduced and results are given for parsing with them. Finally, we compare the experimentally found heads with those based on the linguistic intuition. The paper concludes with future directions of our research.

2 PACE Architecture

The overall architecture of PACE – Parser Comparison and Evaluation platform – has been described in [8].

PACE provides an efficient implementation of standard parser tasks:

- syntactic analysis of natural language sentences based on context free grammars that could be large and highly ambiguous;
- efficient representation of derivation trees;
- pruning of the trees by means of the application of contextual constraints;
- selecting n most probable trees based on the frequency characteristics obtained from treebanks;
- visualization and printing the parsing trees in graphical form.¹

All these above mentioned functions are implemented as plugins that can be modified or even substituted by a better implementation.

The whole process of building final syntactic structures from a given input sentence always requires several steps. Particular components of PACE correspond roughly to the phases of analysis.

¹ The tool was developed by Pavel Rychlý at Faculty of Informatics, Masaryk University, Brno, Czech Republic

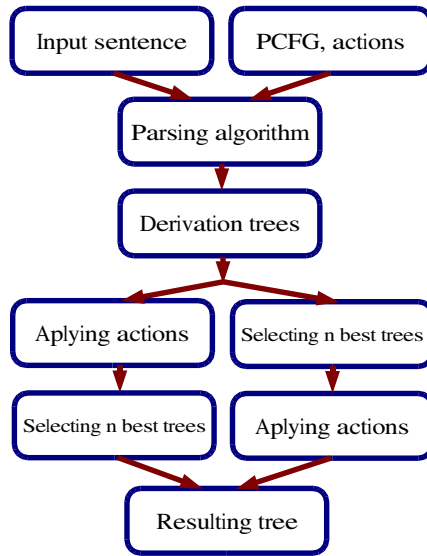


Fig. 1. Modules and data flow in PACE

As mentioned above, if the crucial components of the system are to be freely changeable and not to be fixed in one monolith, one has to pay for the flexibility. In our case, the cost is a postponement of the contextual constraint rule until the whole output structure is constructed, i.e. until all the possible derivations for the given sentence are computed. This means that the system cannot interleave the processing of contextual constraints and the processing of the context-free backbone of the grammar. It has been shown in [9], that by preferring the postponement over the interleaving not a great deal is lost.

Such an approach allows the separation of constraint application functionality which is thus independent of the particular parsing algorithm used. The parsing algorithm can be simply changed, some steps of the parsing process can be omitted or new components added. The users of PACE can even determine the order of analysis steps in some cases. This is demonstrated in Figure 1 – after the completion of the output structure containing all the derivational trees, the user can choose the sequence of the application of contextual constraints first and then the computation of n most probable trees.

PACE provides the parser construction kit independent of particular parsing algorithms. Currently four different parser modules are available – standard bottom-up and top-down chart parsers [10], GLR parser [11] and a special type of head driven chart parser [7].

3 Optimizing Heads for Parsing

The efficiency of head-driven parsing depends to a considerable extent on the choice of grammar rule heads². PACE enables to perform such a resource-demanding set of tests that, to the best of our knowledge, could not be accomplished till now.

3.1 Optimization Procedure

The HDddm parsing technique employed in our experiments was described in [12]. Similarly to [7,13], it improves the process of viable hypotheses confirmation. HDddm (head-driven with dependent dot move) refers to the fact that the move of one “dot” in the head-driven parsing step is dependent on the opposite move of the other one.

The process of finding the positions of rule heads that are optimal for a given parsing algorithm can be summarized as follows: Grammar rules are taken one after the other. The analysis of the given input is run for all possible head positions in the rule. The best head position (the position for which the number of edges in the resulting chart is minimal) is chosen and the rule is given back to the grammar. This is done for all grammar rules.

This “greedy” algorithm finds optimal head positions for given input sentence. It is obvious that one can obtain grammars with different heads for every input sentence. Thus, we choose the most often used head position for every rule and build the final grammar.

We should mention that the grammars created by taking the most frequently used head positions need not to be optimal for the given set of input sentences. One would have to parse all the input data set instead of just one sentence in one step of the optimization process to get an optimal result. However, such a procedure would be harder to parallelize and our experiments on a chosen part of the test set suggest that the output of the simple algorithm is close enough to the optimal solution.

3.2 Configuration of Experiments

We report experiments on two natural language grammars and relevant sets of inputs. The first is the ATIS grammar and the corresponding data set from the pool of common grammars and test data provided at

<http://www.cogs.susx.ac.uk/lab/nlp/carroll/cfg-resources/>.

(These web pages resulted from discussions at the Efficiency in Large Scale Parsing Systems Workshop at COLING 2000, where one of the main conclusions was the need for a bank of data for standardization of parser benchmarking.)

The ATIS grammar consists of 4,592 rules, 192 nonterminals and 357 preterminals. The data set includes 98 sentences of which 71 are grammatical and 27 do not belong to the language generated by the grammar (they have been excluded from the experiments).

We run 1,149,703 analyses (71 sentences \times 4592 grammar rules \times the number of right hand side symbols – 3.52 in average). We divided the input sentences into two sets and run the optimization process in parallel on two Pentium 2.4 GHz workstations. The analyses took 9 hours 23 minutes (total time without initialization, grammar reading, etc.)

² The experimentally chosen heads are called “keys” in [2]

The second series of tests were carried out on the Czech grammar developed for the robust Czech parser [14]. The grammar consists of 2,915 rules, 135 nonterminals and 40 preterminals. The data set includes 100 sentences. All sentences are grammatically correct.

We run 2,469,600 analyses ($100 \text{ sentences} \times 2915 \text{ grammar rules} \times \text{the number of right hand side symbols} - 8.47 \text{ in average}$). The analyses took 8 hours 32 minutes (total time without initialization, grammar reading, etc.) on a Pentium 2.4 GHz workstation.

3.3 Results

Table 1 presents the results of the optimization procedure for the ATIS grammar. The column captioned ATIS gives the parsing time for the baseline – the grammar with heads on the leftmost preterminal. If the rule does not contain any preterminal, than the head is on the leftmost symbol on the right hand side. ATIS-H is the grammar resulting from the optimization described above. ATIS-HS presents characteristics of parsing with the optimal grammars for individual sentences.

Table 1. HDddm parsing on ATIS grammars with different head positions

Grammar	ATIS	ATIS-H	ATIS-HS
Total time (sec)	10.45	3.16	1.30
# edges	1 345 544	398 261	148 866
# edges (optimal parser)	15 252	15 252	15 252
# edges per sentence	13730.04	4063.89	1519.04
Parser optimality	1.13%	3.83%	10.25%
# edges / # edges ATIS	1.00	0.29	0.11

The number of edges for an optimal parser is the value which would be produced by an ideal parsing method that could be able to determine what edges will form the resulting chart. The chart itself (with some additional structures) is employed to represent resulting derivation trees. Thus only edges used in this structure are considered. The “Parser optimality” row shows ratio between the tested and the optimal parser.

It can be seen that parsing with the resulting generated grammar (ATIS-H) is more than three times faster than that with “naive” head positions. The ATIS-HS system is not practical as it uses different grammars for different input sentences. However, the comparison of ATIS-H and ATIS-HS clearly demonstrates that it could be worth looking for smart heuristics to predict the optimal head position of grammar rules based on a particular input.

Table 2 presents results of the optimization procedure for the Czech grammar. Here, we do not need to set the heads for the baseline. The column captioned Czech refers to the results of parsing with the grammar of Czech where the position of rule heads is given by linguistic intuition – the governing nodes are set as heads.

The improvement of the parser performance is not so impressive in the case of the Czech grammar. However, the application of the head optimization algorithm still means a considerable reduction of the time required for parsing. As the baseline grammar employs linguistic heads, we could compare linguistically motivated setting with experimentally found positions of heads.

Table 2. HDddm parsing on Czech grammars with different head positions

Grammar	Czech	Czech-H	Czech-HS
Total time (sec)	4.57	2.20	1.51
# edges	822 989	306 937	191 666
# edges (optimal parser)	76 038	76 038	76 038
# edges per sentence	8229.89	3069.37	1916.66
Parser optimality	9.2%	24.8%	39.7%
# edges / # edges Czech	1.00	0.37	0.23

Table 3. Comparison of linguistically motivated and empirically determined heads

Description	Rule example	Linguistic head	Empirical head
reflexive verb construction	clause \rightarrow subj R VR	VR	R
proper noun phrase	np \rightarrow N np_prop_names	np	np_prop_names
conditional clause	clause \rightarrow condc V intr	V	condc
genitive construction	npl \rightarrow np np_gen	np	np_gen

Table 3 lists the most frequent language phenomena covered by the grammar rules where the position of the linguistic head and the optimized position are different. For example, the linguistic intuition suggest to instantiate the reflexive verb clause (e.g. *Karel se myl* / *Karel washed himself*) from the given verb first. However, the empirical evidence indicates that from the parser optimality’s point of view the best starting point is the reflexive particle in such a case (*se* or *si* in Czech). Similarly for the other language phenomena which are listed in the table, the heads determined in our experiments differ from traditionally defined ones.

4 Conclusions and Future Directions

The above-mentioned results showed that parsing based on grammars with optimized heads can be much more efficient than that with linguistic heads. It is clear that the heuristics suggesting linguistic heads as good indicators for rule instantiation in head-driven approaches should be given up. In accordance with [2], we conclude that the choice of good rule heads for parsing based on a particular grammar is an entirely empirical issue.

The information about linguistic heads still plays an important role in our grammar for Czech. The head-dependent relations are specified in every rule of the metagrammar to enable the dependency graph output [15]. We could add a specification of appropriate heads for parsing as well. However, as PACE enables to select optimal rule head positions efficiently, we are able to run the optimization process for every new version of our grammar. Our future research will therefore focus on full automatization of the process and on further reduction of time complexity of the task. Another interesting direction is the effort to define heuristics that predict the optimal head position of grammar rules based on the particular input.

Acknowledgements

This work was supported by Grant Agency of the Czech Republic Grants GACR 201/99/D028 and GACR 405/03/0913 and by Ministry of Education of the Czech Republic Research Intent CEZ:J07/98:143300003.

References

1. van Noord, G.: An efficient implementation of the head-corner parser. *Computational Linguistics* **23** (1997).
2. Oepen, S., Callmeier, U.: Measure for measure: Parser cross-fertilization – towards increased component comparability and exchange. In: *Proceedings of IWPT 2000, Trento, Italy* (2000) 140–149.
3. Kay, M.: Head driven parsing. In: *Proceedings of International Workshop on Parsing Technologies, Pittsburg* (1989).
4. Bouma, G., van Noord, G.: Head-driven parsing for lexicalist grammars. Experimental results. In: *Proceedings of the 6th Conference of the EACL, Utrecht, The Netherlands* (1993).
5. Smrž, P., Horák, A.: Large scale parsing of Czech. In: *Proceedings of Efficiency in Large-Scale Parsing Systems Workshop, COLING 2000, Saarbrücken: Universitaet des Saarlandes* (2000) 43–50.
6. Moore, R.C.: Improved left-corner chart parsing for large context-free grammars. In: *Proceedings of the 6th IWPT, Trento, Italy* (2000) 171–182.
7. Satta, G., Stock, O.: Head-driven bidirectional parsing: A tabular method. In: *Proceedings of IWPT 1989, Pittsburg* (1989) 43–51.
8. Kadlec, V., Smrž, P.: PACE – parser comparison and evaluation. In: *Proceedings of the 8th International Workshop on Parsing Technologies, IWPT 2003, Le Chesnay Cedex, France, INRIA, Domaine de Voluceau, Rocquencourt* (2003) 211–212.
9. Maxwell III, J.T., Kaplan, R.M.: The interface between phrasal and functional constraints. In: Rosner, M., Rupp, C.J., Johnson, R., (Eds.): *Proceedings of the Workshop on Constraint Propagation, Linguistic Description, and Computation, Instituto Dalle Molle IDSIA, Lugano* (1991) 105–120. Also in *Computational Linguistics*, Vol. 19, No. 4, 571–590, 1994.
10. Sikkel, K.: *Parsing Schemata: A Framework for Specification and Analysis of Parsing Algorithm*. Springer, Berlin (1996).
11. Tomita, M.: *Efficient Parsing for Natural Languages: A Fast Algorithm for Practical Systems*. Kluwer Academic Publishers, Boston, MA (1986).
12. Horák, A., Kadlec, V., Smrž, P.: Enhancing best analysis selection and parser comparison. In: Sojka P. et al. (Eds.): *Text, Speech and Dialogue, Proceedings of the Fifth International Conference TSD 2002, Brno, Czech Republic*, pp. 461–467, Springer-Verlag, LNAI 2448.
13. Sikkel, K., op den Akker, R.: Predictive head-corner parsing. In: *Proceedings of IWPT 1993, Tilburg/Durbuy* (1993) 267–276.
14. Smrž, P., Horák, A.: Implementation of efficient and portable parser for Czech. In: *Text, Speech and Dialogue: Proceedings of the Second International Workshop TSD 1999, Springer-Verlag* (1999) 105–108 *Lecture Notes in Artificial Intelligence* 1692.
15. Smrž, P., Horák, A.: Probabilistic head-driven chart parsing of Czech sentences. In: Sojka P. et al. (Eds.): *Text, Speech and Dialogue, Proceedings of the Third International Conference TSD 2000, Brno, Czech Republic*, pp. 81–86, Springer-Verlag, LNAI 1902.