PACE — PARSER COMPARISON AND EVALUATION

Vladimír Kadlec and Pavel Smrž
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
E-mail: {xkadlec,smrz}@fi.muni.cz

Abstract

The paper introduces PACE — a parser comparison and evaluation system for the syntactic processing of natural languages. The analysis is based on context free grammar with contextual extensions (constraints). The system is able to manage very large and extremely ambiguous CF grammars. It is independent of the parsing algorithm used. The tool can solve the contextual constraints on the resulting CF stucture, select the best parsing trees according to their probabilities, or combine them. We discuss the advantages and disadvantages of our modular design as well as how efficiently it processes the standard evaluation grammars.

There are various parsing algorithms for CF grammars that can differ in many aspects. The attemps to compare the pros and cons of them appeared at the dawn of parser development history and to this day there is a lively interest in comparing and evaluating of natural language parsing methods. Unfortunately, there is no general evaluation procedure acceptable to all researchers and developers. For example, the number of edges in a chart is usually given as a measure to compare chart parsing algorithms. However, as shown in [1], there can be algorithms that do not differ in this respect but the processing time on a given grammar and an input differ considerably.

The primary method of assessing the efficiency of a parsing algorithm is therefore only empirical – one has to compare the time taken to parse a set of test sentences by each particular parser based on a shared grammar. Unfortunately, the proposed methodology calls for standard implementations of reference parsers which has been found to be the most challenging task.

Our work has attempts to add precisely this step to the objective parser comparison – to provide reference implementations of the most popular parsing algorithms and to provide a system with a well-defined interface between its components. Such a design could encourage extensions of the system by other researchers.

The tool is also designed to be efficient across whole process of parsing. The system differs from educational or "toy" systems in this respect as it can be employed in real applications. Efficiency is at the expense of the relative complexity of PACE. The modularity of the system, which is discussed in detail in the following section, seems to help here.

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 [2];
- selecting n most probable trees based on the frequency characteristics obtained from tree-banks;
- visualization and printing the parsing trees in graphical form.[1]

All these above mentioned functions are implemented as plugins that can be modified as needed 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.

As mentioned above, if the crucial components of the system are to be freely changeable and not be fixed in one monolith, one has to pay for the modularity. 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.

Such an approach allows the separation of constraint application functionality which is thus independent on the particular parsing algorithm used. The parsing algorithm can be simply changed, some steps of the parsing process can be ommited or new components added. The users of PACE can even decide the order of analysis steps in some cases. 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.

Note, the system accepts input sentences as sequences of preterminals corresponding to each word. PACE does not offer any special mechanism for lexicon lookup, because the integration of morphological analyser is expected. For example, the system incorporates the morphological analyzer `ajka` (see [3]) when parsing Czech sentences. Of course, it is possible to include the whole lexicon to the grammar in the form of a large set of simple rules, but it could be prohibitively inefficient even for medium-sized lexicons.

References

[1] R. C. Moore. Time as a measure of parsing efficiency. In *Proceedings of Efficiency in Large-Scale Parsing Systems Workshop, COLING'2000*, pages 23–28, Saarbrucken: Universitaet des Saarlandes, 2000.

[2] A. Horák, V. Kadlec, and P Smrž. Enhancing best analysis selection and parser comparison. In *Text, Speech and Dialogue: Proceedings of the 5th International Workshop TSD 2002*, Brno, Czech Republic, 2002. Springer Verlag, Lecture Notes in Artificial Intelligence, Volume 2448.

[3] Radek Sedláček and Pavel Smrž. A New Czech Morphological Analyser *ajka*. In *Text, Speech and Dialogue, 4th International Conference, TSD 2001*, Czech Republic, 2001. Springer-Verlag, Berlin.

[1] The tool was developed by Pavel Rychly at our Faculty