ON PARSING CONTROL FOR EFFICIENT TEXT ANALYSIS

Fabio Ciravegna and Alberto Lavelli
Istituto per la Ricerca Scientifica e Tecnologica
I-38050 Loc. Pantè di Povo, Trento, Italy
e-mail: {cirave|lavelli}@irst.itc.it

1. Introduction

Experience has shown that traditional models of language analysis as used in interfaces are not suitable to analyze texts, as in many cases they tend to visit the entire search space, pursuing every possible solution; moreover when a failure occurs they tend to hypothesize some exotic linguistic phenomena instead of a lack in their own coverage. As a consequence when confronted with complex inputs such as texts, a standard analyzer has to face a huge search space, with consequent excessive waste of computational resources, as well as reduced coverage. In particular coverage has shown to be the crucial aspect: on one hand it is well-known that parsers can not be expected to find full parses for every sentence; on the other hand coverage can increase the combinatorics of parsing and the likelihood of incorrect results. A linguistic analyzer has therefore to be controlled to pursue efficiency and robustness, especially when extensive linguistic resources are provided to reach broad coverage. Efficiency means visiting the search space looking for the most probable solutions, delaying or pruning other possibilities. Robustness means being able to gracefully cope with gaps in the system's knowledge.

In this paper we propose a control strategy for reducing the inefficiency of a broad coverage parser. It uses scores coming from extra-linguistic criteria, e.g text segmentation information, domain-specific heuristics, and the frequency of linguistic phenomena. The analyzer is also able to produce partial results when it is not possible to derive a global analysis, or its search becomes too expensive.

2. Parsing Control Strategies

The main component of the system's linguistic analyzer is an *agenda-based bottom-up bidirectional chart parser* [4], coupled with an agenda management mechanism which sorts tasks and rules to be applied, so to focus the analysis on the most promising solutions.

The keystone of the adopted control mechanism is a preliminary preprocessing phase based on shallow linguistic techniques. This phase is divided in two steps: text segmentation and text classification. Sentence segmentation is a typical technique used in text analysis [3]: in our approach segments represent basic constituents (such as simple nominal, verbal, prepositional phrases, etc.) and are detected via pattern matching. During text classification the current text is assigned to classes through statistical hierarchical pattern matching; the result of text classification is a set of templates to be filled in by the linguistic processor. During linguistic analysis the agenda management uses the preprocessor results to control the parser: the segmentation results are used to split the sentence analysis in two steps (i.e. segment parsing and segment combination), whereas information about the templates is used, among others, as heuristics to sort the tasks in the agenda [1].

During segment parsing the segments produced by the preprocessor are analyzed, producing basic constituents such as simple NPs, PPs, and so on. This means that the combination of edges crossing the boundaries of segments is prevented (i.e. delayed until segment combination) by an appropriate setting of the control strategy. At this level no other control mechanism is applied, as the number of edges generated by such basic constituents is quite small.

During segment combination the syntactic analysis is completed considering all the tasks in the agenda (included those that cross the boundaries of segments). This is the very moment when the control mechanism is necessary. The analyzer is controlled through the sorting of the tasks in the agenda; four different criteria are employed to contribute the scores used in task ranking:

- *Segmentation*: the tasks generated by edges spanning a whole segment are given an extra score. This criterion introduces some kind of top-down control on the parser actions. It is

useful for example to avoid some phenomena similar to garden paths that are easily followed by an uncontrolled parser.

- *Rule Score:* more frequent rules are applied first; this is particularly important in a free word order language like Italian, where the number of rules seldom applied tends to be very high.
- *Template Filling:* solutions providing information for target template(s) are given an extra score, favoring the most interesting solutions from an applicative point of view.
- *Width:* tasks combining edges spanning wider parts of the input are preferred.

Each criterion produces a numerical score; these four scores are weighted and summed up. The effectiveness of the control strategy relies therefore on the weights given to the different scores. We are currently experimenting different criteria of composition [2]. Some preliminary results seem to indicate that segmentation is to be given the maximum weight, followed by template filling, rule score and width. This strategy seems to be appealing from an intuitive point of view too, as the main role is played by the top-down control for avoiding garden paths, followed by the goal-driven strategy (i.e. template filling); the rule score weight is still to be determined precisely.

At the end of parsing the (possible) different (complete or partial) solutions produced by the linguistic analyzer can be evaluated according to the same four criteria (i.e. segmentation, template filling, rule score and width). In this case the most important one will be of course the template filling criterion.

The strategy mentioned above allows the parser to focus on the most promising solutions. Nevertheless focusing is not enough for efficiency, unless coupled with a process of reduction of the search space; in our case that reduction means pruning some tasks from the agenda. Unfortunately, pruning is dangerous, as it can prevent the parser from finding the correct solution. In our approach the agenda is pruned of some low score tasks when at least a "satisfying" solution has been found. Some experiments have been carried out to determine a criterion for considering a solution as "satisfying"; currently the following minimal criterion is used: when an edge spanning all the sentence has been found that has maximum segmentation score and an *acceptable* template score, the tasks with lower segmentation scores are pruned. This strategy also allows to overcome possible segmentation errors [2].

3. Parser at Work

The modules described so far have been integrated in a system for text understanding currently under development at IRST [1]. The system has been implemented in Common Lisp. As for the linguistic modules of the system, both syntactic and semantic information are encoded using a formalism based on Typed Feature Structures.

We have been experimenting on a corpus composed of short news (average 70 words) taken from the Italian financial newspaper "IL SOLE 24 Ore". Up to now, we have carried on only some preliminary experiments that seem to show a certain reduction in the amount of edges built by the parser, i.e. between 10% and 40%. The computational cost of the application of the control strategies is negligible.

In the near future we plan to do more extensive experiments to find the most effective way of combining the four criteria and to consider additional criteria. An evolution of the agenda pruning strategy using also rule scores is under study.

References

[1] Fabio Ciravegna and Nicola Cancedda. Integrating shallow and linguistic techniques for information extraction from text. In *Proceedings of the Fourth Congress of the Italian Association for Artificial Intelligence*, Firenze, Italy, October 1995. To be published in Lecture Notes in Artificial Intelligence, Springer-Verlag.

[2] Fabio Ciravegna and Alberto Lavelli. Controlling bidirectional parsing for efficient text analysis. Technical Report 9508-02, IRST, July 1995.

[3] Paul S. Jacobs. To parse or not to parse: Relation-driven text skimming. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, pages 194–198, Helsinki, Finland, 1990.

[4] G. Satta and O. Stock. Formal properties and implementation of bidirectional charts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1480–1485, Detroit, MI, 1989.