Grammar Organization for Cascade-Based Parsing in Information Extraction

Fabio Ciravegna **Alberto Lavelli**
ITC-irst Centro per la Ricerca Scientifica e Tecnologica
via Sommarive 18, 38050 Povo (TN) ITALY
{cirave|lavelli}@irst.itc.it

Recently there has been an increasing interest in finite-state techniques for NLP. Finite-state transducers have been used for a number of NLP tasks, from morphological analysis and shallow parsing to semantic processing. In [3] we have proposed a parsing approach for Information Extraction (IE) based on finite-state cascades. The parser aims at building a syntactic representation equivalent (from an IE point of view) to the ones built by systems based on full parsing. The approach is inspired by the work of Abney on finite-state cascades for parsing unrestricted text [1] and provides: (i) a method for efficiently and effectively performing full parsing on texts for IE purposes; (ii) a way of organizing generic grammars that simplifies changes, insertion of new rules and integration of domain-oriented rules. Parsing is based on the application of a fixed sequence of cascades of rules. It is performed deterministically in three steps: (1) chunking; (2) clause recognition and nesting; (3) modifier attachment.

The approach has been developed as part of LE-FACILE, a successfully completed EU project for multilingual text classification and IE [4]. The approach is currently used in Pinocchio, an environment for developing and running IE applications [2]. The proposed approach has been tested mainly for Italian, but proved to work also for English and partially for Russian.

Due to space limitations, in this paper we concentrate exclusively on the issues connected with grammar organization. For further details on the adopted formalism, the parsing approach, and some experimental results, see [3].

Rules are grouped into **cascades** that are finite, ordered sequences of rules. Cascades represent elementary logical units, in the sense that all the rules in a cascade deal with some specific construction (e.g., subcategorization of verbs). From a functional point of view a cascade is composed of three segments: s**1** contains rules that deal with idiosyncratic cases for the construction at hand; s**2** contains rules dealing with the regular cases; s**3** contains default rules that fire only when no other rule can be successfully applied.

The initial generic grammars for chunking and clause recognition are designed to cover the most frequent phenomena in a restrictive sense. Additional rules can be added to the grammar (when necessary) for coping with the uncovered phenomena, especially domain-specific idiosyncratic forms. The limited size of the grammar makes modifications simple (e.g., our clause recognition grammar for Italian is composed of 66 rules).

The deterministic approach combined with the use of cascades and segments makes grammar modifications simple, as changes in a cascade (e.g., rule addition/modification) influence only the following part of the cascade or the following cascades. This makes the writing and debugging of grammars easier than in approaches based on context-free grammars, where changes to a rule can in principle influence the application of any rule in the grammar.

The grammar organization in cascades and segments allows a clean definition of the grammar parts. Each cascade copes with a specific phenomenon (modularity of the grammar). All the rules for the specific phenomenon are grouped together and are easy to check.

The segment structure of cascades is suitable for coping with the idiosyncratic phenomena of restricted corpora. As a matter of fact domain-oriented corpora can differ from the standard use of language (such as those found in generic corpora) in two ways: (i) in the frequency of the constructions for a specific phenomenon; (ii) in presenting different (idiosyncratic) constructions. Coping with different frequency distributions is conceptually easy by using deterministic parsing and cascades of rules, as it is sufficient to change the rule order within the cascade coping with the specific phenomenon, so that more frequently applied rules are first in the cascade. Coping with idiosyncratic constructions requires the addition of new rules.

Finally from the point of view of grammar organization, defining segments brings some added value to ordered cascades. Generic rules (in s2) are separated from domain specific ones (in s1); rules covering standard situations (in s2) are separated from recovery rules (in s3). In s2, rules are generic and deal with unmarked cases. In principle s2 and s3 are units portable across the applications without changes. Domain-dependent rules are grouped together in s1 and are the resources the application developer works on for adapting the grammar to the specific corpus needs (e.g., coping with idiosyncratic cases). Such rules generally use contexts and/or introduce domain-dependent (semantic) constraints in order to limit their application to well defined cases. S1 rules are applied before the standard rules and then idiosyncratic constructions have precedence with respect to standard forms.

Segments also help in parsing robustly. S3 deals with unexpected situations, i.e. cases that could prevent the parser from continuing. For example the presence of unknown words is coped with after chunking by a cascade trying to guess the word's lexical class. If every strategy fails, a recovery rule includes the unknown word in the immediately preceding chunk so to let the parser continue. Recovery rules are applied only when rules in s1 and s2 do not fire.

References

[1] Abney S. Partial parsing via finite-state cascades. In *Proceedings of the ESSLI '96 Robust Parsing Workshop*, 1996. Also available at the URL `http://www.sfs.nphil.uni-tuebingen.de/~abney/Papers.html`.

[2] Ciravegna F. and A. Lavelli. The Pinocchio Information Extraction Toolkit. `http://ecate.itc.it:1024/projects/pinocchio.html`.

[3] Ciravegna F. and A. Lavelli. Full text parsing using cascades of rules: An information extraction perspective. In *Proceedings of EACL99*, Bergen, Norway, 1999.

[4] Ciravegna, F., Lavelli, A., Mana, M., Gilardoni, L., Mazza, S., Ferraro, M., Matiasek, J., Black, W. J., Rinaldi, F. and D. Mowatt. FACILE: Classifying texts integrating pattern matching and information extraction. In *Proceedings of IJCAI99*, Stockholm, Sweden, 1999.