A PRACTICAL DEPENDENCY PARSER

Vincenzo Lombardo
Leonardo Lesmo
Dipartimento di Informatica - Università di Torino - c.so Svizzera, 185 - 10149 Torino - Italy
Centro di Scienze Cognitive - Università di Torino - via Lagrange, 3 - 10123 Torino - Italy
e-mail: {vincenzo, lesmo}@di.unito.it

The working assumption is that cognitive modeling of NLP and engineering solutions to free text parsing can converge to optimal parsing. The claim of the paper is that the methodology to achieve such a result is to develop a concrete environment with a flexible parser, that allows the testing of various psycholinguistic strategies on real texts. In this paper we outline a flexible parser based on a dependency grammar.

Cognitive modeling of human parsing and engineering solutions to text parsing are usually far apart. The goal of this research is to test whether the two fields can converge to optimal parsing. Efficiency in NLP is mostly obtained by separating the two phases of parsing and interpretation: an active chart parser can produce an annotated compact representation of the exponential number of syntactic trees in cubic time; the interpreter reads out the syntactic trees and builds a semantic representation for each valid one. The annotation corresponds to associate a disjunction of tuples with an edge. The interpretation process is theoretically exponential, and various authors have tried to reduce the computation time by means of devices that extend the disjoint representation to feature constraints application (Maxwell, Kaplan 1993) and to the semantic interpretation (Rim et al. 1990), or annotate the nodes of a shared-packed forest with the calls to the interpretation routines, that are "lazily" executed only when needed (Harper 1994).

Interleaving syntax and semantics has been proposed mostly in cognitive approaches (Schubert 1984) (Crain, Steedman 1985) (Hirst 1987). Since it is unreasonable to think that the human being carries on the exponential number of alternative paths that arise during the parsing process, many authors have proposed that many ambiguities are solved with the contribution of other sources, like semantics and context. Incremental interpretation, as this approach is usually referred to, involves an interleaving of the modules that can be implemented at various degrees. Interleaving requires the distinction of the several paths during the parsing process: only a limited number of paths is allowed for continuation (active paths), while many others are abandoned (inactive paths). The whole process is theoretically exponential, and we cannot isolate a part that can be executed in a polynomial time. The approach followed in this paper is to give up the polynomiality of parsing, by providing a method to factorize the paths in an efficient way and to easily switch between active and inactive paths when a recovery phase is required. We have implemented a flexible parser for testing various psycholinguistic hypotheses on the human parsing mechanism and we have equipped the parser with a recovery mechanism that allows an intelligent backtracking (Lombardo 1995). Currently we are testing the practical validity of a number of heuristics on an Italian corpus.

The syntactic representation is a *dependency graph*, that compacts all the dependency trees associated with a sentence. The dependency graph in the upper part fig. 1 is associated with the sentence "I saw a tall old man in the park with a telescope".

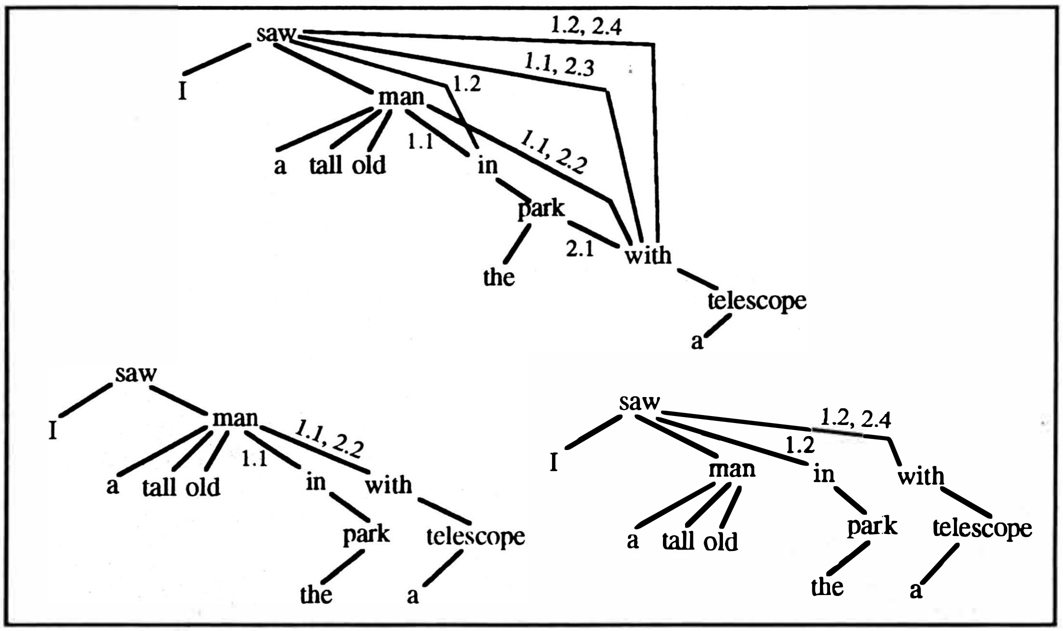

Figure 1. The dependency graph for the sentence "I saw a tall old man in the park with a telescope"

Various paths are identified by labelling some parts of the dependency graph with *indices* during the parsing process. An index is a pair of natural numbers h.k. Two indices, $h_1.k_1$ and $h_2.k_2$, are *compatible* iff either $h_1 \neq h_2$ or $h_1 = h_2$ & $k_1 = k_2$. A *path* is a set of indices compatible with each other. For each dependency tree T of a sentence there exists one and only one path P in G such that the tree consisting of all the vertices and edges reachable through P from a root vertex is equal to T. The bottom of fig. 1 shows the two dependency trees associated with the paths {1.1, 2.2}, {1.2, 2.4}.

The parser combines top-down predictions with bottom-up filtering. In case of ambiguity the parser explores in parallel the several paths for an input fragment. The dependency graph built in this phase keeps the paths distinct by means of the indices introduced above. The parser associates an integer h with each point of ambiguity and an integer h.k with each solution for the ambiguity h. Then the parser chooses the best path according to some heuristic. This becomes the *active path*, that is used for continuation; the others are the *inactive paths*. The parser supports an incremental interpretation at a fine degree of interleaving because the syntactic structure is always connected and each operation of the parser modifies the structure. The path representation is the communication medium between the parser and the recovery mechanism when something goes wrong.

References

Crain S., Steedman M., *On not Being Led Up the Garden Path: The Use of Context by the Psychological Syntax Processor*, in Dowty, Karttunen, Zwicky (eds.): **Natural Language Parsing**, Cambridge Univ. Press, 1985, pp. 320-358.

Harper M. P., *Storing Logical Form in a Shared-Packed Forest*, **Computational Linguistics 20**, 1994, pp. 649-660.

Hirst G., **Semantic Interpretation and the Resolution of Ambiguity**, Cambridge Univ. Press, 1987.

Lombardo V., *Parsing and Recovery*, Proc. 17th Annual Conference of the Cognitive Science Society, Pittsburgh, 1995. pp.648-653.

Maxwell J. T., Kaplan R. M.: The Interface between Phrasal and Functional Constraints, **Computational Linguistics 19**, 1993, pp. 571-590.

Rim H. C., Seo J., Simmons R. F., *Transforming syntactic graphs into semantic graphs*, Proc. of the 28th Annual Meeting of the ACL, Pittsburgh (PA), 1990, pp. 47-53.

Schubert L. K., *On Parsing Preferences*, Proc. COLING 84, Stanford, 1984, pp. 247-250.