Comments on Joshi

Lauri Karttunen
Rank Xerox Research Centre
6 Chemin de Maupertuis
F-38240 Meylan
France
`lauri.karttunen@Grenoble.RXRC.Xerox.com`

There is currently a great deal of interest among computational linguists in robust, wide-coverage parsing methods. There is a popular misconception that this trend is a recent innovation brought upon us by the availability of large quantities of text in electronic form. In fact, as the paper by Joshi shows, it goes back to the very beginnings of computational linguistics. The goal was the same then as it is now and so were the methods of getting there.

It is fascinating to see, how many of the currently popular techniques for robust parsing are already present, fully articulated in the 1959 Univac parser from UPenn. Among its remarkable features are

- multiword tokens ("because of", "in front of")
- tagging words by ambiguity classes ("cool V/A")
- rule-based disambiguation
- syntactic markup by finite-state transduction
- depth-first strategy with backtracking and pushdown store for the analysis of recursive structures
- default selection of one structure among alternative analyses with option for later revision

Although the Upenn parser has had a great influence on other systems, most notably, the string grammar prgram at NYU, some its features have faded from the collective consciousness only to be reinvented by a new generation of computational linguists. I will comment here briefly on the handling of ambiguities and the marking of first-order constituents (elementary noun, adjunct, and verb phrases).

If a word is unambiguous, the lexicon of the UPenn parser assigns to it a simple part-of-speech tag. Ambiguous words initially get a tag that shows the alternative analyses, such as V/A ("cool"), N/V ("study"), N/V/A ("total"), etc. The system tries to select the appropriate unambiguous tag the basis of the surrounding context by means of rules that are remarkably similar to the ones that show up in systems such as the TAGGIT system (Greene and Rubin , 1971) and the Helsinki Constraint Grammar (Karlsson et al, 1994). In essence, each disambiguation rule is a finite-state transducer that replaces an ambiguous tag by a less ambiguous one depending on the neighboring tags. One can now easily compile such transducers from expressions such as

```
N/V´ -> N  || _ P=of

``Select V if the preposition `of´ follows,
where V´ marks verbs that do not take `of´
as part of their object (``consist of")."

N/V -> N, N/V/A -> N/A || T1 | T2 _

``Eliminate the V possibility immediately
after a T1 (`a´, `an´) or T2 (`the´).
```

Some of the disambiguation rules check an unbounded amount of context to the left or to the right in the sentence. For example, the rule that disregards the verb possibility after an adjective,

```
N/V -> N || CNTX A _
```

sorts our the contexts where the adjective cannot be the head of the noun phrase. For example, it chooses the N tag in cases such as "a wise study", "this wise study", and "from wise study" but leaves the ambiguity unresolved in cases like "the wise study".

Another interesting aspect of the disambiguation module is that the rules are applied in sequence starting with the rules that are least likely to produce errors and the process is iterated as long as there is some change. This same reach for a fixed-point we also see in later systems such as Roche (1994).

The syntactic markup is also done in a sequence. The system first marks the boundaries of elementary noun phases, then prepositional phrases, and finally simple verb phrases. For each type of constituent, the alternate structures are described by a tree. Elementary noun phrases are described from right to left, the others from left-to-right, reflecting the position of the head.

The tree graph constitutes a simple automaton for recognizing the longest possible instance of the pattern. The procedure that matches the input against the tree inserts brackets to mark the beginning and the end of each constituents. Thus it interprets the graph as a transducer. It inserts an opening bracket as soon as it detects the beginning of the constituent, scans ahead one symbol at the time until it reaches something that does not belong and inserts the closing bracket just in front of that last symbol. The principle of longest match is built in the graph.

© 1996 L. Karttunen
Proceedings of the ECAI 96 Workshop
Extended Finite State Models of Language
Edited by A. Kornai.

There is no reentrancy in the tree but in effect it might just as well be a cyclic graph because the interpreter returns to a node already traversed when it encounters another word with the same category. Thus the noun phrase tree allows any number of determiners and adjectives.

The result of the initial syntactic analysis is a relatively flat structure with the elementary noun phrases nested within the PPs and VPs, similar to the output of current "chunking" systems, for example Abney (1991).

There has been a tremendous amount of progress since the late 50s in the computing hardware. We have also advanced is some other aspects. Finite-state transducers for syntactic parsing can now be compiled easily from regular expressions in a way that incorporates the directionality and longest match constraints in the structure of the network itself (Karttunen 1996). Such transducers can also be composed to construct a single transducer that has the same effect as the original transducers have when applied in a sequence.

On the other hand, the fact that the antique parser from 1959 strikes us as so modern in other respects, is a sad reminder that the progress in the area of linguistic analysis and in the art of grammar construction has been much less spectacular.