

Ambiguity resolution in a reductionistic parser *

Pasi Tapanainen & Atro Voutilainen

Research Unit for Computational Linguistics

P.O. Box 4 (Keskuskatu 8)

FIN-00014 University of Helsinki

Finland

1 Introduction

We are concerned with grammar-based surface-syntactic analysis of running text. Morphological and syntactic analysis is here based on tags that express surface-syntactic relations between functional categories such as Subject, Modifier, Main verb etc.; consider the following simple sentence:

I	PRON	◊SUBJECT
see	V PRES	◊MAINVERB
a	ART	◊>N
bird	N	◊OBJECT
.		FULLSTOP

2 Description of the parsing system

The parsing system consists of the following modules:

2.1 Preprocessor

The preprocessor normalises the input text, detects sentence boundaries and punctuation marks, and identifies idioms and other fixed syntagms.

2.2 Morphological analyser

The ENGTWOL morphological analyser is a 55,000 entry Koskeniemi-style morphological description of English that assigns all recognised input word forms with all possible morphological readings as a disjunctive list.

Those words not recognised by the ENGTWOL analyser are analysed by a heuristic module; part-of-speech readings are assigned on the basis of the form of the word (endings etc.).

The morphologically analysed sentences are enriched with syntactic and word boundary ambiguities and converted into regular expressions by simple awk programs.

2.3 Finite-State parser

The Finite-State parser transforms sentences and rules into finite-state automata. The parser computes the intersection of the sentence automaton and all rule automata; the intersection is the parse of the sentence.

The grammar also contains a heuristic section that can be used to rank multiple analyses.

*The lexicon is adopted from the ENGCG parser that has been supported by TEKES, the Finnish Technological Development Center, and the work on Finite-state syntax has been partly supported by the Academy of Finland.

3 Sample analysis

The sentence *Its leadership was insulted by editors* gets two analyses, when no heuristics are applied:

it	<NonMod>	PRON	GEN	SG3	◊>N	◊◊
leadership	N	NOM	SG		◊SUBJ	◊
be	V	PAST	SG1.OR.3	VFIN	◊AUX	◊
insult	<SVO>	PCP2			◊MV MAINC	◊◊
by	PREP				◊ADVL	◊
editor	N	NOM	PL		◊P<	◊
.		FULLSTOP				◊◊

it	<NonMod>	PRON	GEN	SG3	◊>N	◊◊
leadership	N	NOM	SG		◊SUBJ	◊
be	<SV>	<SVC/N>	<SVC/A>			
	V	PAST	SG1.OR.3	VFIN	◊MV MAINC	◊
insult	PCP2				◊SC	◊
by	PREP				◊ADVL/N<	◊
editor	N	NOM	PL		◊P<	◊
.		FULLSTOP				◊◊

Syntactic tags

◊>N	determiner or premodifier
◊SUBJ	subject of a finite clause
◊AUX	auxiliary in a finite clause
◊MV	main verb in a finite clause
MAINC	finite main clause
◊ADVL	adverbial
◊P<	preposition complement
◊SC	subject complement
◊ADVL/N<	adverbial or a postmodifier of a nominal

References

- [Voutilainen and Tapanainen, 1993] Atro Voutilainen and Pasi Tapanainen. Ambiguity resolution in a reductionist parser In *Proceedings of EACL-93*. Utrecht, Netherlands, 1993.