A Cascaded Finite-State Parser for Syntactic Analysis of Swedish

Dimitrios Kokkinakis and Sofie Johansson Kokkinakis
Department of Swedish/Språkdata
Box 200, SE-405 30
Göteborg University, Göteborg
SWEDEN
{svedk,svesj}@svenska.gu.se

Abstract

This report describes the development of a parsing system for written Swedish and is focused on a grammar, the main component of the system, semi-automatically extracted from corpora. A cascaded, finite-state algorithm is applied to the grammar in which the input contains coarse-grained semantic class information, and the output produced reflects not only the syntactic structure of the input, but grammatical functions as well. The grammar has been tested on a variety of random samples of different text genres, achieving precision and recall of 94.62% and 91.92% respectively, and average crossing rate of 0.04, when evaluated against manually disambiguated, annotated texts.

1 Introduction

This report describes a parsing system for fast and accurate analysis of large bodies of written Swedish. The grammar has been implemented in a modular fashion as finite-state, cascaded machines, henceforth called *Cass-SWE*, a name adopted from the parser used, *Cascaded analysis of syntactic structure*, (Abney, 1996). Cass-SWE operates on part-of-speech annotated texts and is coupled with a pre-processing mechanism, which distinguishes thousands of phrasal verbs, idioms, and multi-word expressions. Cass-SWE is designed in such a way that semantic information, inherited by named-entity (NE) identification software, is taken under consideration; and grammatical functions are extracted heuristically using finite-state transducers. The grammar has been manually acquired from open-source texts by observing legitimately adjacent, part-of-speech chains, and how and which function words signal boundaries between phrasal constituents and clauses.

2 Background

2.1 Cascaded Finite-State Automata

Finite-state technology has had a great impact on a variety of Natural Language Processing applications, as well as in industrial and academic Language Engineering. Attractive properties, such as conceptual simplicity, flexibility, and space and time efficiency, have motivated researchers to create grammars for natural language using finite-state methods: Koskenniemi *et al.* (1992); Appelt *et al.* (1993); Roche (1996); Roche & Schabes (1997). The cascaded, finite-state mechanism we use in this work is described in Abney (1997):

> *"...a finite-state cascade consists of a sequence of strata, each stratum being defined by a set of regular-expression patterns for recognizing phrases. [...] The output of stratum 0 consists of parts of speech. The patterns at level l are applied to the output of level l-1 in the manner of a lexical analyzer [...] longest match is selected (ties being resolved in favour of the first pattern listed), the matched input symbols are consumed from the input, the category of the matched pattern is produced as output, and the cycle repeats...*", (p. 130).

2.2 Swedish Finite-State Grammars

There have been few attempts in the past to model Swedish grammars using finite-state methods. K. Church at MIT implemented a Swedish, regular-expression grammar, inspired by ideas from Ejerhed & Church (1983). Unfortunately, the lexicon and the rules were designed to parse a very limited set of sentences. In Ejerhed (1985), a very

general description of Swedish grammar was presented. Its algorithmic details were unclear, and we are unaware of any descriptions in the literature of large scale applications or implementations of the models presented. It seems to us that Swedish language researchers are satisfied with the description and, apparently, the implementation on a small scale of finite-state methods for noun phrases only, (Cooper, 1984; Rauch, 1993). However, large scale grammars for Swedish do exist, employing other approaches to parsing, either radically different, such as the *Swedish Core Language Engine*, (Gambäck & Rayner, 1992), or slightly different, such as the *Swedish Constraint Grammar*, (Birn, 1998).

2.3 Pre-Processing

By pre-processing we mean: (i) the recognition of multi-word tokens, phrasal verbs and idioms; (ii) sentence segmentation; (iii) part-of-speech tagging using Brill's (1994) part-of-speech tagger, and the EAGLES tagset for Swedish, (Johansson-Kokkinakis & Kokkinakis, 1996). The general accuracy of the tagger is at the 96% level, (98,7% for the evaluation presented in table (1)). Tagging errors do not influence critically the performance of Cass-SWE[1] (*cf.* Voutilainen, 1998); (iv) semantic inheritance in the form of NE labels: *time sequences, locations, persons, organizations, communication and transportation means, money expressions* and *body-part*. The recognition is performed using finite-state recognizers based on trigger words, typical contexts, and typical predicates associated with the entities. The performance of the NE recognition for Swedish is 97.4% precision, and 93.5% recall, tested within the AVENTINUS[2] domain. Cass-SWE has been integrated in the *General Architecture for Text Engineering* (GATE), Cunningham *et al.* (1996).

3 The Grammar Framework

The Swedish grammar has been semi-automatically extracted from written text corpora by observing two phenomena: (i) which part-of-speech *n-grams*, are not allowed to be adjacent to each other in a constituent, and (ii) how and which function words signal boundaries between phrases and clauses. (i) uses the *Mutual Information*, statistics, based on the n-grams. Low n-gram frequencies, such as verb/noun-determiner, gave reliable cues for clause boundary, while high values such as numeral-noun did not, and thus rejected. Observation (i) is related to the notion of *distituent grammars*, "*...a distituent grammar is a list of tag pairs which cannot be adjacent within a constituent...*", Magerman & Marcus (1990); (ii) is a supplement of (i), which recognizes formal indicators of subordination/co-ordination, such as conjunctions, subjunctions, and punctuation.

3.1 Syntactic Labelling and the Underlying Corpus

The syntactic analysis is completed through the recognition of a variety of phrasal constituents, sentential clauses, and subclauses. We follow the proposal defined by the EAGLES (1996), *Syntactic Annotation Group*, which recognizes a number of syntactic, metasymbolic categories that are subsumed in most current categories of constituency-based syntactic annotation. The labelled bracketing consists of the syntactic category of the phrasal constituent enclosed between brackets. Unlabelled bracketing is only adopted in cases of unrecognized syntactic constructions. The corpora we used consisted of a variety of different sources, about 200,000 tokens, collected in AVENTINUS. The rules are divided into levels, with each level consisting of groups of *patterns* ordered according to their internal complexity and length. A pattern consists of a *category* and a *regular expression*. The regular expressions are translated into *finite-state automata*, and the union of the automata yields a single, deterministic, finite-state, level recognizer, (Abney, 1996). Moreover, there is also the possibility of grouping words and/or part-of-speech tags using morphological and semantic criteria.

3.2 Grammar Rules

Some of the most important groups include:

- **Noun Phrases, Grammar$_0$**: the number of patterns in grammar$_0$ is 180, divided in six different groups, depending on the length and complexity of the patterns. A large number of (parallel) coordination rules are also implemented at this level, depending on the similarity of the conjuncts with respect to several different characteristics, (*cf.* Nagao, 1992).
- **Prepositional Phrases, Grammar$_1$**: the majority of prepositional phrases are noun

[1] The parser can be tolerant of the erroneous annotation returned by the tagger, e.g. in the distinction between Swedish adjective-participles in (-t). This is accomplished by constructing rules that contain either adjective or participle in the following manner:

`np → ARTICLE(ADJECTIVE|PARTICIPLE) NOUN`

[2] AVENTINUS (LE-2238), *Advanced Information System for Multilingual Drug Enforcement.* (http://svenska.gu.se/aventinus)

phrases preceded by a preposition. *Trapped* adverbials, belonging to the noun phrase and not identified while applying grammar_0, are merged within the np. Both simple and multi-word prepositions are used.

- **Verbal Groups, Grammar_2**: identifies and labels phrasal, non-phrasal, and complex verbal formations. The rules allow for any number of auxiliary verbs, possible intervening adverbs, and end with a main verb or particle. A distinction is made between finite/infinite active/passive verbal groups.

- **Clauses, Grammar_3 and Grammar_4**: the clause resolution is based on surface criteria, outlined at the beginning of this chapter, and the rather fixed word order of Swedish. Grammar_3 distinguishes different types of subordinate clauses; while Grammar_4 recognizes main clauses. A unique level is designated for each type of clause

3.3 Grammatical Functions

Grammatical functions are heuristically recognized using the topographical scheme, originally developed for Danish, in which the relative position of all functional elements in the clause is mapped in the sentence, (Diderichsen, 1966).

3.4 An Example

The following short example illustrates the input and output to Cass-SWE:

'Under 1998 gick 8 799 företag i konkurs i Sverige.', i.e. '*During 1998, 8 799 companies went bankrupt in Sweden.*'

The input to Cass-SWE is an annotated version of the text:

```
'Under/S 1998/MC/tim gick/VMISA 8_799/MC
företag/NCN(SP)NI/org i/S konkurs/NCUSNI
i/S Sverige/NP/lcg./F'.
```

Output:

```
[main_clause
  TIME=[rp head=Under sem=tim
    [S head=Under sem=n/a Under]
    [np head=1998 sem=tim
      [MC head=1998 sem=tim 1998]]]
  [vg-active-finite head=gick sem=n/a
    [VMISA head=gick sem=n/a gick]]
  SUBJ=[np head=företag sem=org
    [MC head=8_799 sem=n/a 8_799]
    [NCN(SP)NI head=företag sem=org företag]]
  P-OBJ=[pp head=i sem=n/a
    [S head=i sem=n/a i]
    [np head=konkurs sem=n/a
      [NCUSNI head=konkurs sem=n/a konkurs]]]
  [pp head=i sem=lcg
    [S head=i sem=n/a i]
    [np head=Sverige sem=lcg
      [NP head=Sverige sem=lcg Sverige]]]
  [F .]]
```

Here S: preposition; MC: numeral; VMISA: finite, active verb; NCUSNI/NCN(SP)NI: common nouns; NP: proper noun and F: punctuation; while tim: time sequence; org: organization and lcg: geographical location. The output produced reflects the coarse-grained semantics and part-of-speech used in the input, as well as the head of each phrase and the grammatical functions: TIME, SUBJ(ect) and P-OBJ(ect).

4 Evaluation

The performance of the parser partly depends on the output of the tagger and the rest of the pre-processing software. Our way of dealing with how "correct" the performance of the parser is, follows a practical, pragmatic approach, based on consultation of modern Swedish syntax literature. We use the metrics: precision (P), recall (R), F-value (F) and cross-bracketed rate. $F = (\beta^2+1)\,PR/\beta^2 P+R$, where β is a parameter encoding the relative importance of (R) and (P); here β=1. Evaluation is performed automatically using the *evalb* evaluation software, (Sekine & Collins, 1997).

4.1 'Gold Standard' and Error Analysis

For the evaluation of Cass-SWE we use three types of texts: (i) a sample taken from a manually annotated Swedish corpus of 100,000 words with grammatical information (*SynTag*, Järborg, 1990); (ii) newspaper material; and (iii) a test suite, for non-common constructions, by consulting Swedish syntax literature. Texts (ii) and (iii) were annotated manually. The total number of tokens was 1,500 and sentences 117.

The evaluation results are given in Table (1), for both noun phrases (NPs), and full chunk parsing (All). The errors found can be divided into: (i)

Table 1: Cass-SWE, Performance

	P	R	F	Cross
NPs	97.82%	94.52%	96.17%	0.03
All	94.62%	91.92%	93.2%7	0.04

errors in the texts themselves, which we cannot control and are difficult to discover if the texts are not proofread prior to processing; (ii) errors produced by the tagger; and (iii) grammatical errors produced by the parser, caused mainly by the lack of an appropriate pattern in the rules, and almost exclusively in higher order clauses due to

structural ambiguity and coordination problems. None of the errors in (i) and (ii) have been manually corrected. This was a conscious choice, so that the evaluation of the parsing will be based on unrestricted data.

5 Conclusion

We have described the implementation of a large coverage parser for Swedish, following the cascaded finite-state approach. Our main guidance towards the grammar development was the observation of how and which function words behave as delimiters between different phrases, as well as which other part-of-speech tags are not allowed to be adjacent within a constituent. Cass-SWE operates on part-of-speech annotated texts using coarse-grained semantic information, and produces output that reflects this information as well as grammatical functions in the output. A corpus, annotated syntactically, is a rich source of information which we intend to use for a number of applications, e.g. information extraction; an intermediate step in the extraction of lexical semantic information; making valency lexicons more comprehensive by extracting sub-categorization information, and syntactic relations.

References

Abney, S. 1996. Partial Parsing via Finite-State Cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*, Prague, Czech Rep.

Abney, S. 1997. Part-of-Speech Tagging and Partial Parsing, In *Corpus-Based Methods in Language and Speech Processing*, Young S. and Bloothooft G., editors, Kluwer Acad. Publishers, Chap. 4, pp. 118-136.

Appelt, D.E., J. Hobbs, J. Bear, D. Israel, and M. Tyson. 1993. FASTUS: A Finite-State Processor for Information Extraction from Real-World Text, In *Proceedings of the IJCAI '93*, France.

Birn, J. 1998. *Swedish Constraint Grammar*, Lingsoft Inc., Finland, *forthcoming.*

Brill, E. 1994. Some Advances In Rule-Based Part of Speech Tagging, In *Proceedings of the 12th AAAI '94*, Seattle, Washington.

Cooper, R. 1984. Svenska nominalfraser och kontext-fri grammatik, In *Nordic Journal of Linguistics*, Vol. 7:115-144, (in Swedish).

Cunningham, H., R. Gaizauskas, and Y. Wilks. 1995. *A General Architecture for Text Engineering (GATE) - A New Approach to Language Engineering R&D*, Technical report CS-95-21, University of Sheffield, UK.

Diderichsen, P. 1966. *Helhed og Struktur*, G.E.C. GADS Forlag, (in Danish).

EAGLES. 1996. *Expert Advisory Group for Language Engineering Standards*, EAG-TCWG-SASG/1.8 , http://www.ilc.pi.cnr.it/EAGLES/home.html. Visited 01/08/1998.

Ejerhed, E. and Church, K. 1983. Finite State Parsing, In *Papers from the 7th Scandinavian Conference of Linguistics*, Karlsson F., editor, University of Helsinki, Publ. No. 10(2):410-431.

Ejerhed, E. 1985. En ytstruktur grammatik för svenska, In *Svenskans Beskrivning 15*, Allén, S., L-G. Andersson, J. Löfström, K. Nordenstam, and B. Ralph, editors, Göteborg, (in Swedish).

Gambäck, B. and Rayner, M. 1992. *The Swedish Core Language Engine*, CRC-025, http://www.cam.sri.com. Visited 01/10/1998.

Johansson-Kokkinakis, S. and Kokkinakis, D. 1996. *Rule-Based Tagging in Språkbanken*, Research Reports from the Department of Swedish, Göteborg University, GU-ISS-96-5.

Järborg, J. 1990. *Användning av SynTag*, Research Reports from the Department of Swedish, Göteborg University, (in Swedish).

Koskenniemi, K., P. Tapanainen, and A. Voutilainen. 1992. Compiling and Using Finite -State Syntactic Rules, In *Proceedings of COLING '92*, Nantes, France, Vol. 1:156-162.

Magerman, D.M. and Marcus, M.P. 1990. Parsing a Natural Language Using Mutual Information Statistics, In *Proceedings of AAAI '90*, Boston, Massachusetts.

Nagao, M. 1992. Are the Grammars so far Developed Appropriate to Recognize the Real Structure of a Sentence?, In *Proceedings of 4th TMI*, Montréal, Canada, pp. 127-137.

Rauch, B. 1993. Automatisk igenkänning av nominalfraser i löpande text, In *Proceedings of the 9th NODALIDA*, Eklund, R., editor, pp. 207-215, (in Swedish).

Roche, E. 1996. *Parsing with Finite-State Transducers*, http://www.merl-com/reports/TR96-30. Visited 12/03/99.

Roche, E. and Schabes, Y., editors, 1997. *Finite-State Language Processing*, MIT Press.

Sekine, S. and Collins, M.J. 1997. *The evalb Software*, http://cs.nyu.edu/cs/projects/proteus/evalb. Visited 14/12/97.

Voutilainen, A. 1998. Does Tagging Help Parsing? A Case Study on Finite State Parsing, In *Proceedings of the FSMNLP '98*, Ankara, Turkey.