First results of developing a Swedish parser based on a wide coverage grammar

Malin Ahlberg
Center of Language Technology, Gothenburg University, Sweden

This paper describes ~~an~~ on-going work of a rule-based, open source parser for Swedish. The central component will be a wide-coverage grammar implemented in the ~~grammar~~ formalism Grammatical Framework (GF). The resulting parser will be the first deep Swedish parser based on a grammar and will give full syntactic analysis of the input sentences. In addition to GF, we use two other main resources~~;~~ the Swedish treebank Talbanken and the electronic lexicon SALDO. The grammar will be highly reusable, chosen parts of the it ~~can~~ be used for applications dealing with controlled or free natural language. GF also provides libraries for compiling grammars to a number of programming languages.

1 Introduction

Swedish is a North Germanic language spoken by approximately 10 million people. It is a SVO language, and although the word order is relatively strict, inverted order is commonly used to indicate questions or to emphazise different parts of the sentence.
Language technology for Swedish is an area of much interesting research. Our goal is to implement a wide coverage grammar and parser for Swedish. We thereby investigate how the grammar formalism Grammatical Framework can be used for open domain parsing. By starting from an already existing Swedish grammar written in GF, we get an fundamental description of the language. The framework also provides tools such as parsing, generation and a well tested interpretation of the parse trees. Furthermore, there are tools for using the grammar in a number of programming languages.

From the GF grammar we aim to implement a robust parser. The parser will be evaluated both by experts and by comparing the results of parsing sentences in an often used Swedish treebank, Talbanken.

This paper will briefly introduce GF, Talbanken and SALDO in section 2 and describe our current work in section 3.

2 Background

2.1 Grammatical Framework

Grammatical Framework[11] (GF) is a grammar formalism based on functional programming, designed for multilingual grammar applications.

The key idea is to divide the grammar into abstract and concrete parts. The abstract grammar gives a logical representation of the semantics, modeled as abstract trees. Complications occurring in natural languages, such as agreement, case and word order, are abstracted away. That is, the abstract grammar contains no actual linguistic information, but information about what categories are used and how to combine them into trees.

Submitted to:
LREC2012

To add a new language~~s~~ means to give a concrete grammar. This implements the abstract grammar and describes how to translate abstract trees to strings of the language. The framework enables us to parse strings into abstract trees as well as linearize trees into strings. Since all languages supported in an application share~~s~~ a common abstract syntax, we get multilinguality for free.

The grammar acts as an independent module and may be used in different projects. Reusability is further supported by the separation between resource grammars and applic~~i~~ation grammars. The resource grammar~~s is a~~ library provided with ~~the~~ GF ~~package~~, where information about the morphological and syntactical rules for more than 20 different languages ~~are implemented~~. Hence a writer of a application grammar can start her work at a higher level and does not need to describe how to form standard sentences, phrases or decline words.

GF has so far been used in a number of projects, MOLTO[1], TALK[9] and WebAlt[3] to mention a few. All those are special domain applications, dealing with controlled natural language. This project experiments on other usage of a GF grammar - to use it for open domain language.

2.2 Talbanken

Talbanken[4] is a Swedish treebank put together in the 1970s at Lund University. In 2005 it was modernized by Nivre, Nilsson and Hall[10] and enriched with annotation for a full phrase structure analysis. We are using the treebank for development and evaluation.

Although Talbanken contains both written and spoken Swedish, only the prose material, consisting of 6316 sentences will be used in this project. This part was also used when training the data-driven parser Maltparser [7].

2.3 Saldo

SALDO[2] is an open source lexicon resource based on Svenskt Associationslexikon (SAL). It is developed at Språkbanken at Gothenburg University and intended for usage in language technology research. From SALDO, a large GF lexicon has earlier been extracted, containing 50 000 entries. By updating this technique, we would like to reimport SALDO, which has since then been developed. The importing method should be fast and reliable enough to be allow us to always have a fresh version of the dictionary in GF.

3 Work in progress

3.1 A tool for lexical acquisition

To enlarge the lexicon, a tool for automatically acquisition has been created. It has been tested on verbs with good results. It makes use of the *smart paradigm* given in the Swedish resource grammar. The smart paradigm acts as functions that given one form of a word can infer which paradigm it most likely belong to. The smart paradigm for verbs accepts words in present tense indicative form. If needed, it also accepts more verb forms showing the correct inflection.

By combining this method with the information from the tags in Talbanken, the tool interactively generates GF lexicons. Given a list of words, it iteratively tries to figure out how to conjugate each of them. If several forms of a word is given, the program will try to identify the one that carries the most linguistic information, put this in a form recognized by the smart paradigm and ask GF to output a table

with the resulting inflection ~~table~~. If this table contains all other forms from the input list, the program will ask the user to validate the claimed conjunction. The user may now either allow the word to be added to the lexicon, remove it, or demand the program to make another guess. Although using simple techniques, the tool manage to correctly guess 70-75% of the given lemmas when tested on verbs.
For further extractions, there are also earlier developed tools such as Extract[5] for supervised lexicon extraction and FM[6] for programming lexical resources.

3.2 Mapping of trees

The information from the tags in Talbanken can be used for many purposes. We are currently working on an automatic transformation of Talbanken trees to trees in GF format. The translation makes use of the POS tags as well as the syntactic information and the mapping has so far turned out to be unambiguous.

The information given by this mapping may be used for the lexical extraction tools. Those can be enhanced if they are given more data about which form a word is currently used in. The translated trees will also enable us to extract possibilities for how often different functions are used, a feature that would enable disambiguation. Furthermore, the translation makes it easy to identify grammatical constructions missing from the GF grammar and shows how the GF analysis differs from the one made in Talbanken.

Another important use of the mapping is evaluation. ~~By~~ comparing the trees from the parser and from the transformer, ~~we get~~ an interesting way of accomplishing an evaluation.

3.3 Development of the grammar

For multilingual GF applications, translation should always be possible, and therefore the resource abstract may only contain constructions that are common to all implemented languages. Therefore, the resources have to be general and non language specific. Language specific constructs such as stylistic changes, idioms, informal expressions etc may be given in a special module. This module, together with the Swedish resource grammar has been the starting point of this project.

Since many of the languages in the GF library reassemble each other grammatically, they can share much of their grammar implementations. This is usually done by using a `Functor`, which lets a number of languages share parts of their implementation. In addition to simply avoiding code duplication, this technique aids the code maintenance. For Swedish, about 85% of the resource code is shared with the other Scandinavian languages. However, if we aim for a deeper and more covering analysis of Swedish, the implementation of the languages needs to be more separated. A number of grammatical constructions have been added the the Swedish grammar. Those include constructs that are generally not present in other languages, such as the reflexive pronoun *sitt*:
*Han såg **sitt** hus* (*He saw his (own) house*) as opposed to
*Han såg **hans** hus* (*He saw his (an other person's house*)
The possibility to put a ~~part~~ of a sentence in focus has also been added:
Glad var han inte. (*Happy was he not*).
This sort of rephrasing which has little effect on the logical representation is commonly not given by the resource grammar.

In order to keep the grammar ~~as~~ clean and to keep it from allowing syntactical errors, much care is taken to develop this as neatly as possible.

4 Related work

There are other parser for Swedish ~~parser~~ such as the ~~the~~ statistical MaltParser[7] which was trained on Talbanken. CassSwe (Kokkinakis and Johansson-Kokkinakis, 1999) is ~~a~~ based on finite state cascades, whereas shallow parser GTA (Knutsson, Bigert and Kann, 2003) relies on rules of a context free grammar. Both GTA and CassSwe operate~~s~~ on POS tagged text.

5 Evaluation

Compared to a statistical parser which operates like a black box, a rule-based one is not only theoretically interesting, but could also give a more explanatory output since the rules are given their names by a human and hence act~~s~~ as informative labels. Chosen parts of the parser or grammar may also be used in other applications, which may be dealing with controlled natural languages. Automatic language generation from the grammar is also provided by GF.

We intend to evaluate the parser both automatically - by comparing the output the translated trees from Talbanken - and manually by an expert in Swedish grammar.

6 Future Work

After having developed the grammar and lexicon, we plan to make the parser robust by using techniques such as chunk parsing, named entity recognition, methods for handling unknown grammatical constructions such as idioms and ellipses.

It is already possible to add probabilities for GF functions, which serve~~s~~ as a ranking when several parse trees are possible. By adding dependency probabilities, we would like to improve the disambiguation.

7 Conclusion

We intend to implement a robust deep parser for Swedish, by first developing a large scale grammar and then equip this with named entity recognition, a voluminous lexicon, statistical information for disambiguation, a method of parallel chunk parsing etc. So far we have been working on the grammar, the lexical resources and a translation of the treebank Talbanken to GF. The usage of Grammatical Framework gives us the advent age to start from a well-defined system of describing grammar, as well as tools for using the grammar in combination with programming languages like Haskell and Java.

8 Acknowledgments

The work has been funded by Center of Language Technology. I would also like to give special thanks to Aarne Ranta, Ramona Enache, Elisabet Engdahl, Krasimir Angelov and Lars Borin for their help and support.

References

[1] (2010): *MOLTO - Multilingual On-line Translation*.

[2] L. Borin, M. Forsberg & L. Lönngren (2008): *SALDO 1.0 (Svenskt associationslexikon version 2)*.

[3] Olga Caprotti: *WebALT! Deliver Mathematics Everywhere*.

[4] Jan Einarsson (1976): *Talbankens skriftspråkskonkordans*.

[5] Markus Forsberg (2007): *The Extract Tool* Available at `http://www.cs.chalmers.se/~markus/Extract_Tech_Report.pdf`.

[6] Markus Forsberg (2007): *The Functional Morphology Library* Available at `http://www.cs.chalmers.se/~markus/FM_Tech_Report.pdf`.

[7] Johan Hall: *A Hybrid Constituency-Dependency Parser for Swedish*.

[8] Ola Knutsson, Johnny Bigert & Viggo Kann (2003): *A robust shallow parser for Swedish*. In: *In Proc. 14th Nordic Conf. on Computational Linguistics*.

[9] Peter Ljunglof, Bjorn Bringert, Robin Cooper, Ann-Charlotte Forslund, David Hjelm, Rebecca Jonson & Aarne Ranta (2005): *The TALK Grammar Library: an Integration of GF with TrindiKit* Available at `http://www.talk-project.org/fileadmin/talk/publications_public/deliverables_public/TK_D1-1.pdf`.

[10] Joakim Nivre, Jens Nilsson & Johan Hall: *Talbanken05: A Swedish treebank with phrase structure and dependency annotation*. In: *In Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006*, pp. 24–26.

[11] Aarne Ranta (2011): *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford. ISBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth).