Combining Language Resources Into A Grammar-Driven Swedish Parser

Malin Ahlberg, Ramona Enache

Department of Computer Science & Engineering
Gothenburg University
Box 8718, SE-402 75 Gothenburg, Sweden
ahlberg.malin@gmail.com, enache@chalmers.se

Abstract

This paper describes work on a rule-based, open-source parser for Swedish. The central component is a wide-coverage grammar implemented in the GF formalism (Grammatical Framework), a dependently typed grammar formalism based on Martin-Löf type theory. GF has strong support for multilinguality and has so far been used successfully for controlled languages (Angelov and Ranta, 2009) and recent experiments have showed that it is also possible to use the framework for parsing unrestricted language. In addition to GF, we use two other main resources: the Swedish treebank Talbanken and the electronic lexicon SALDO. By combining the grammar with a lexicon extracted from SALDO we obtain a parser accepting all sentences described by the given rules. We develop and test this on examples from Talbanken. The resulting parser gives a full syntactic analysis of the input sentences. It will be highly reusable, freely available, and as GF provides libraries for compiling grammars to a number of programming languages, chosen parts of the the grammar may be used in various NLP applications.

Keywords: GF, Swedish, Parsing

1. Introduction

Our goal is to implement a wide-coverage grammar and parser for Swedish using the GF formalism, and thereby investigate how GF can be used for open-domain parsing. We compile a large-scale grammar to a parser and combine it with the extensive lexicon SALDO. The Swedish treebank Talbanken provides manually tagged trees which we use for improving and evaluating the grammar. The parser will additionally be evaluated by an expert.

We chose our resources so that both our grammar and parser can be freely available and open-source. The target language is Swedish, a North-Germanic language closely related to Norwegian and Danish. The languages share most of their grammatical structures and are mutually intelligible. Swedish is also one of the official languages in Finland and altogether spoken by approximately 9 million people. Swedish syntax is often similar to English, but the morphology is richer and the word order slightly more intricate. It is a verb-second language: the second constituent of a declarative main clause must consist of a verb. The first constituent of the clause is usually made up of the subject, although it likewise could consist of adverbial phrases or objects. Fronting the finite verb marks questions.

This paper will briefly introduce GF, Talbanken and SALDO in Section 2. Section 4. explains how we extract a lexicon, section 5. how we translate Talbanken to GF annotation and section 6. presents the work on the grammar implementation.

2. Background

2.1. Grammatical Framework

Grammatical Framework (Ranta, 2011) is a grammar formalism based on functional programming.

The key idea is to divide a grammar into abstract and concrete parts. The abstract grammar gives a logical representation of the semantics, modeled as abstract trees. The concrete grammars tell how to translate the abstract trees to a given language, and deal with issues such as word order, case and agreement. The framework enables us to parse strings into abstract trees as well as linearize trees into strings.

The grammar acts as an independent module and reusability is further supported by the separation between resource grammars and application grammars. The resource library provided with GF implements morphological and syntactical rules for more than 20 different languages. Hence the writer of an application grammar can start her work at a higher level and does not need to describe how to form standard sentences, phrases or inflect words.

Since many of the languages in the GF library resemble each other grammatically, they can share much of their implementations. This is usually done by using a `Functor` by which we avoid code duplication and which aids the code maintenance.

GF has so far been used in a number of projects, MOLTO[1], TALK (Ljunglöf et al., 2005) and WebAlt (Caprotti, 2006) to mention a few. All those are special domain applications, dealing with controlled natural language. This project takes a different approach by using GF for open domain language, similar to the recently conducted work on translation of patents (España-Bonet et al., 2011).

Using the Swedish resource grammar as our starting point, we get a basic description of the language. The framework provides tools such as parsing, generation and a well-tested interpretation of the parse trees. Furthermore, there are tools for using GF grammars in a number of programming languages like Haskell and Java.

2.2. Talbanken

For development and evaluation, we use the Swedish treebank Talbanken (Einarsson, 1976). It was assembled

[1] http://www.molto-project.eu/

in the 1970s at Lund University and later enriched with annotation for a full phrase structure analysis (Nivre et al., 2006). Although Talbanken contains both written and spoken Swedish, only the prose material, consisting of 6316 sentences, is used in this project. This part was also used when training the data-driven parser Maltparser (Hall et al., 2007).

2.3. SALDO

SALDO (Lars Borin, 2008) is an open source lexicon resource based on Svenskt Associationslexikon. It is developed at Språkbanken at Gothenburg University and intended for usage in language technology research. We have developed tools for extracting GF lexicons from SALDO, described in section 4.

3. Related work

Many years of research have lead to many interesting language technology tools for Swedish. An example is the well-known data-driven Maltparser (Hall et al., 2007), trained on Talbanken. There are also a number of grammar-based parsers, although none is freely available. The cascaded finite state parser CassSwe (Kokkinakis and Kokkinakis, 1999) and The Swedish Constraint Grammar (Birn, 1998) give syntactic analyses. Swedish FDG (Voultanien,2001) uses the Functional Dependency Grammar (Tapanainen and Järvinen, 1997), an extension of the Constraint Grammar formalism, and produces a dependency structure focusing on finding the nominal arguments.

The LinGO Grammar Matrix (Bender et al., 2002), is a starter-kit for building Head-Driven Phrase Structure Grammars (Pollard and Sag, 1994) (HPSG) providing compatibility with tools for parsing, evaluation, semantic representations etc. Translation is supported by using Minimal Recursion Semantics (Copestake et al., 2005) as an interlingua. There is a collection of grammars implemented in this framework, giving broad-coverage descriptions of English, Japanese and German. The Scandinavian Grammar Matrix (Søgaard and Haugereid, 2005) covers common parts of Scandinavian, while Norsource (Hellan and Haugereid, 2003) describes Norwegian. A Swedish version was based upon this by Ahrenberg, covering the morphology and some differences between Swedish and Norwegian. Further, there is the BiTSE grammar (Stymne, 2006), also implemented using the Lingo Matrix, which focuses on describing and translating verb frames.

The Swedish version of the Core Language Engine (CLE) (Gambäck, 1997) gives a full syntactic analysis as well as semantics represented in 'Quasi logical form'. A translation to English was implemented and the work was further developed in the spoken language translator (Rayner et al., 2000). Unfortunately, it is no longer available.

In the TAG formalism (Joshi, 1975), there are projects in getting open-source, wide-coverage grammars for English and Korean, but, to our knowledge, not for Swedish.

The ParGram (Butt et al., 2002) project aims at making wide coverage grammars using the Lexical Functional Grammar approach (Bresnan, 1982). The grammars are implemented in parallel in order to coordinate the analyses of different languages and there are now grammars for English, German, Japanese and Norwegian.

Extract (Forsberg, 2007) is a tool for lexicon extraction compatible with GF, sharing its basic ideas with our lexical acquisition tool. Extract does however not consider parts-of-speech and our tool is developed closer to GF, while still having additional mechanisms for robustness and human support for accuracy.

4. Extracting a large lexicon

The lexicon provided with the GF resources is far too small for open-domain parsing. This section describes the process of importing SALDO, which is compatible with GF, and easily translated to GF format. As SALDO is continuously updated, the importing process has been designed to be fast and stable enough to be redone at any time.

4.1. Implementation

The basic algorithm for importing SALDO was implemented by Angelov (2008) and produces code for a GF lexicon. For each word in SALDO, it decides which forms should be used as input to the GF smart paradigms. The smart paradigm is a function which given one form of a word, can infer which paradigm it most likely belongs to. For verbs, this will in most cases mean giving the present tense form, see figure 1.

```
mkV "knyter" ;
```

Figure 1: First code produced for the verb *knyta* ('tie')

All assumed paradigms are printed to a temporary lexicon, which will produce an inflection table for every entry when compiled. The tables are compared to the information given in SALDO and if the tables are equal the code for the word is saved. If the table is erroneous, another try is made by giving more forms to the smart paradigm. For example 1, the smart paradigm will fail to calculate the correct inflection table. In the next try both the present and the past tense are given:

```
mkV "knyter" "knöt" ;
```

Figure 2: Second output for the verb *knyta*

The program is run iteratively until the GF table matches the one given in SALDO, or until there are no more ways of using the smart paradigm. The verb *knyta* will need three forms:

```
mkV "knyter" "knöt" "knutit"
```

Figure 3: Final output for the verb *knyta*

4.2. Results

The resulting dictionary contains more than 100 000 entries, approximately 80 % of the total size of SALDO. There are a number of reasons why some words were not imported, the most obvious one is that we do not want all categories from SALDO in the GF lexicon. Prepositions, numerals, personal pronouns etc. are assumed to be present in the resource grammars and should not be added again. SALDO contains many pronouns which are not analyzed the same way in GF. Before adding them to our lexicon, we need to do more analyzing to find their correct GF-category. Categories involving multiple words are usually handled as idioms and should be given in a separate lexicon. In total six types of words were considered for the extraction:

	SALDO/GF	Example
Adverb	**ab/Adv**	*ofta* ('often')
Adjective	**av/A**	*gul* ('yellow')
Noun	**nn/N**	*hus* ('house')
Verb	**vb/V**	*springa* ('run')
Reflexive verbs	**vbm/V**	*raka sig* ('shave')
Particle verbs	**vbm/V**	*piggna till* ('perk up')

Figure 4: Word classes imported from SALDO

Most but not all words of these categories have been imported. One reason why the importing phase would fail is that SALDO, unlike GF, only contains the actually used word forms. For technical reasons, the smart paradigm might need forms never used. Consider for example the plural tantum noun *glasögon* ('glasses'). The smart paradigm requires a singular form, and since the program could not find this in SALDO, there was no way of adding the lemma to the lexicon. When the program failed to import a noun, this was often the explanation. Words of this type may be added manually, for *glasögon* we could use the ostensibly correct singular form *glasöga*, although this has another meaning ('glass-eye'). The same problem occurred for the irregular s-verbs (*synas* ('show') or *umgås* ('socialize')) which made up 61.5 % of the failing verbs of type `vb`.

In a few cases the smart paradigms could not generate the correct declination.

When testing the coverage of Talbanken, we found that there are around 2500 word forms still missing, excluding the ones tagged as names and numbers. This number may seem very high, but 4/5 of the word forms are compounds and when performing the intended parsing, an additional analysis identifying compounds is preformed before looking-up the words in the lexicon. Talbanken also contains a small number of spelling errors, which probably are enumerated among our missing words. The majority of the missing words are only used once.

A list of words that were given different labels in GF than in Talbanken has been composed, consisting of about 1600 entries. Many of those are acceptable and reflects the difference made in the analyses, while others are examples of words that are still missing from the lexicon.

Missing words	$\sim$ 2500 word-forms
Ignoring compounds	$\sim$ 500 word-forms
Used more than once	$\sim$ 500 word forms
Used more than once, ignoring compounds	$\sim$ 150 word-forms

Figure 5: Number of Talbanken words still missing

Valency information, which is crucial for GF, is not given in SALDO and hence not in the imported lexicon. Instead we are working on methods of extracting this from Lexin[2].

4.3. A tool for lexical acquisition

As we extract our main lexicon from SALDO, we have also created a tool for semi-automatic acquisition to complement the lexicon. Like the SALDO importer it makes use of the smart paradigm given in the Swedish resource grammar. Unlike the SALDO importer, this tool does not require any particular word form, but operates on any verb form and iteratively tries to figure out how to conjugate each of them. If several forms of a word are 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. If the table contains all other conjugations from the input list, the program will ask the user to validate the claimed paradigm. This step is needed since the input may not provide information enough to automatically do the validation. The user may now either allow the word to be added to the lexicon, remove it or request another guess. The user hence only needs to decide if each paradigm is correct or not.

The tool has been tested on verbs. Although using simple techniques it manages to assign the correct paradigm to 70-75 % of the given lemmas. A smaller test has shown that out of the accepted lemmas, the correct guess is made directly in 75% of the cases, whereas the user has to reject one or more guesses for 25%.

5. Mapping of Talbanken trees

The information from the tags in Talbanken can be used for many purposes. We have developed 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.

Figure 6 shows an example of a visualized Talbanken05 tree of the sentence *"Katten på bilen blir större"* ("The cat on the car gets bigger") and its translation to GF.

The translation gives us means to evaluate our parser. By both parsing a Talbanken sentence and transforming its annotated tree, we can easily inspect if the results are equal. Additionally, the mapping shows which grammatical constructions that are still missing from the GF grammar and shows how the GF analysis differs from the one made in Talbanken. If there are words missing from our dictionary, the rich POS-tags may help us to automatically find the correct declination and add it to the lexicon. Further, our parser will need probabilities of how often a function is used. The GF treebank we achieve from the translation is a

[2] http://spraakbanken.gu.se/lexin/

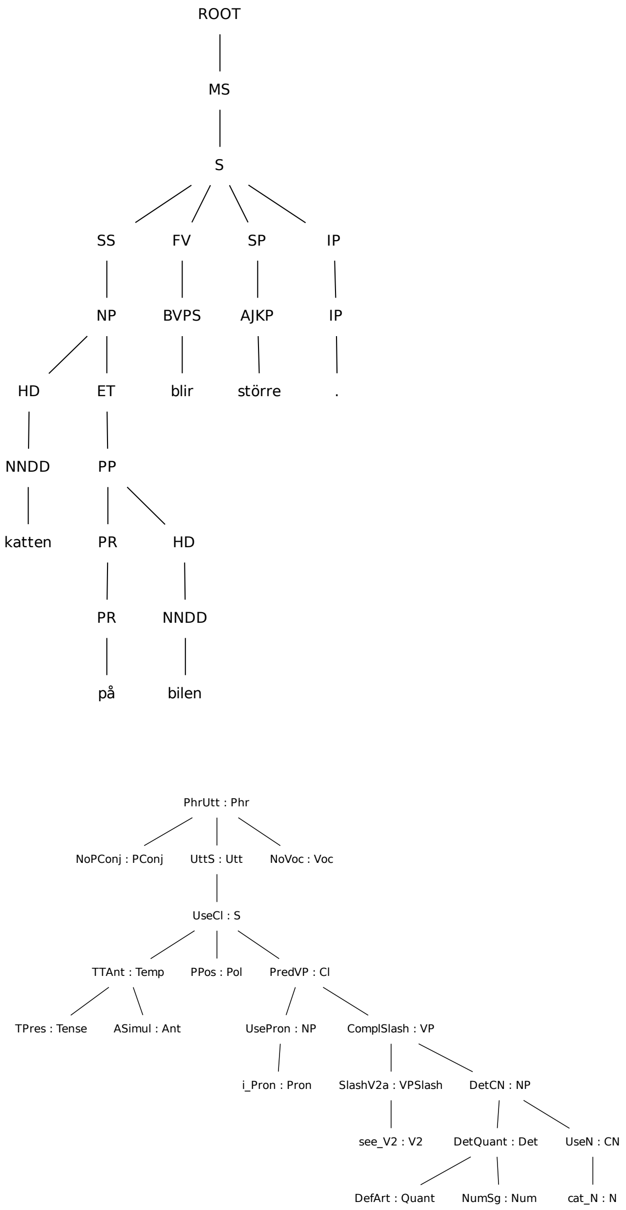

Figure 6: The Talbanken tree and the GF tree for the sentence *"Katten på bilen blir större"*.

good source for this information.

5.1. Results and evaluation

The development was mostly example-driven, and at least one rule for the translation of every Talbanken-tag has been implemented. Shorter sentences, with less than 10 words, have been prioritized in order to get a coverage of the most fundamental constructions.

When evaluating the mapping, the results strongly depend on which restrictions we put on the input. One of the reasons why a node cannot be translated, is the use of the tags show in figure 7. The **PU** tag is used for graphic listings, and not for fluent text. In our grammar there is naturally no corresponding function; the listings are meant for making the text look nice in folders etc and are outside the scope for the grammar itself. The tags **XX** and **NAC** are often used since Talbanken makes a difference between subject and object noun phrases. The analysis of elliptical expression in (1)

(1) För stora krav.
"Too high demands."

contains the tags **XX** and **NAC**, since it is not obvious whether the noun phrase is used as subject or an object. The tags shown in figure 7 occur quite frequently in the treebank and are always translated to metas, which lowers our result.

NAC	Not a constituent
XP	Other (non-coordinated) phrase
DB	Doubled function
PU	List item
XX	Unclassifiable part-of-speech

Figure 7: Untranslatable tags

The main goal has been to be able to translate shorter sentences, with no idioms or conjunction. If we assure that the lexicon contains the correct word class for all lemmas involved, we can restore more than 85 % of the nodes in the original tree. If we lift all the restrictions excluding the PU, we get 65 % coverage. If we test randomly collected sentences that do not contain any of the tags listed in figure 7, 72 % can be restored (see figure 8)

No list items	65 %
No special punctuation or bad tags	72 %
Short sentences with known words	85 %

Figure 8: Number of nodes in each translated tree not put to meta

6. Development of the grammar

An important part of this project has been to develop the Swedish GF grammar and to adapt it to cover constructions used in Talbanken. As a grammar implementation can never be expected to give full coverage of a language, we aim for a grammar fragment which gives a deep analysis of the most important Swedish constructions. The starting point has been the GF resource grammar and the new implementation is still compatible with this.

For Swedish, about 85% of the GF resource code is shared with the other Scandinavian languages. However, if we aim for a deeper and more comprehensive analysis of Swedish, the implementation of the languages needs to be more independent. The resource grammar gives a good start and our present grammar covers constructions such as declarative sentences, questions, passives, imperatives, relative clauses, cleft constructions etc. A number of constructs that are generally not present in other languages and therefore not given by the resources, have also been added. These include the use of the reflexive pronoun *sitt*:

*Han såg **sitt** hus* ("He saw SELF's house")
as opposed to
*Han såg **hans** hus* ("He saw his (another person's) house")
Fronting words or phrases is very common in Swedish and are now allowed by the grammar:
***Glad** var han inte.* ("**Happy** was he not").
This sort of rephrasing is not given by the resource grammar, since it has little effect on the logical representation.

7. Evaluation and Future Work

The project has so far resulted in

- a large-scale GF lexicon and a program to redo the importation when needed
- an extended grammar covering an important part of Swedish
- a comparison and translation between GF and another annotation

Besides being capable of reimporting SALDO, the lexicon extraction program could also be modified for importing other lexical resources. The only requirement is that the resource provides inflection tables.

The grammar has been extended and enhanced, and its current status is a specialized extension of the resource grammar. Besides parsing, the grammar may well be used for language generation. By the renewed import of SALDO, we have doubled the size of the lexicon and thereby added many of the commonly used words that were missing from the older version. This is of course a big improvement but the lexical part still requires some work before it can be made good use of. The lexicon is too big to use with the current techniques as its size exhausts the current incremental parsing algorithm. However, new research is being conducted to improve the GF run-time system.

When it comes to parsing, we do not get far without robustness. The grammar in itself is by no means robust, and just one unexpected punctuation mark, unknown word or ellipsis will cause the parsing of the whole sentence to fail. Parsing bigger parts of Talbanken would hence give very low results at this stage, and a comparison of the results would not be of much value as there would not be enough material to do be able to do any interesting analysis.

We are currently working on chunk parsing by which we get robustness and a possibility to limit the memory usage. We perform disambiguation by using probabilities extracted from our translation of Talbanken. We combine this with simple named entity recognition and compounding analysis. For evaluation, we intend to use 10 % of Talbanken, chosen so that is does not infer with our test data.

The result will be evaluated both automatically – by comparing the output of the translated trees from Talbanken – and manually by professor Elisabet Engdahl[3]. She also evaluates the intermediate results.

8. Conclusion

We have developed the main components for a deep Swedish parser; an extended grammar and lexicon and material for evaluation and disambiguation. By starting from the GF resource grammar, we got a well-defined system for describing language. We have developed tools for extending the lexicon with words from Talbanken, and we show how to make use of the information in the manually tagged treebank. The usage of GF allows us to start from a well-defined system for describing grammar, as well as tools for parsing. All parts of the project are open-source and may thus be used in other applications. The grammar and the lexicon may be beneficial also when working with controlled languages, as it increases the coverage of the Swedish resource grammar.

9. Acknowledgments

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

10. References

Krasimir Angelov and Aarne Ranta. 2009. Implementing Controlled Languages in GF. In *CNL*, pages 82–101.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14.

Juhani Birn. 1998. Swedish Constraint Grammar. Technical report, Lingsoft Inc.

J. Bresnan. 1982. *The Mental Representation of Grammatical Relations*. MIT Press.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar project. In *COLING-02 on Grammar Engineering and Evaluation*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Olga Caprotti. 2006. WebALT! Deliver Mathematics Everywhere.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics – An Introduction. *Research on Language and Computation*, 3:281–332.

Jan Einarsson. 1976. Talbankens skriftspråkskonkordans. Lund University: Department of Scandinavian Languages.

Cristina España-Bonet, Ramona Enache, Adam Slaski, Aarne Ranta, Lluís Marquez, and Meritxell Gonzalez. 2011. Patent translation within the MOLTO project. In *Proceedings of the 4th Workshop on Patent Translation, MT Summit XIII*, pages 70–78.

Markus Forsberg. 2007. *Three Tools for Language Processing: BNF Converter, Functional Morphology, and Extract*. Ph.D. thesis, Göteborg University and Chalmers University of Technology.

[3] http://svenska.gu.se/om-oss/personal/elisabet-engdahl

Björn Gambäck. 1997. *Processing Swedish Sentences: A Unification-Based Grammar and Some Applications*. Ph.D. thesis, The Royal Institute of Technology and Stockholm University, Dept. of Computer and Systems Sciences.

Johan Hall, Joakim Nivre, and Jens Nilsson. 2007. A Hybrid Constituency-Dependency Parser for Swedish. In *In Proceedings of NODALIDA–2007*, pages 284–287.

Lars Hellan and Petter Haugereid. 2003. The NorSource Grammar - an excercise in the Matrix Grammar building design. In *Proceedings of Workshop on Ideas and Strategies for Multilingual Grammar Engineering*.

Aravind K. Joshi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences archive*.

Dimitrios Kokkinakis and Sofie Johansson Kokkinakis. 1999. A Cascaded Finite-State Parser for Syntactic Analysis of Swedish. In *In Proceedings of the 9th EACL*, pages 245–248.

Lennart Lönngren Lars Borin, Markus Forsberg. 2008. The hunting of the BLARK - SALDO, a freely available lexical database for Swedish language technology. *Resourceful language technology. Festschrift in honor of Anna Sågvall Hein*, pages 21–32.

Peter Ljunglöf, Björn Bringert, Robin Cooper, Ann-Charlotte Forslund, David Hjelm, Rebecca Jonson, and Aarne Ranta. 2005. The talk grammar library: an integration of gf with trindikit.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *In Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006)*, pages 24–26.

C. Pollard and I. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Aarne Ranta. 2011. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford.

Manny Rayner, David Carter, Pierrette Bouillon, Vassilisi Digalakis, and Mats Wirén. 2000. *The spoken language translator*. Cambridge Univerisy press.

Anders Søgaard and Petter Haugereid. 2005. A brief documentation of a computational HPSG grammar specifying (most of) the common subset of linguistic types for Danish, Norwegian and Swedish. *Nordisk Sprogteknologi 2004*, pages 247–56.

Sara Stymne. 2006. Swedish-English Verb Frame Divergences in a Bilingual Head-driven Phrase Structure Grammar for Machine Translation. Master's thesis, Linköping University.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *In Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71.