Dependency-based Sentence Simplification for Increasing Deep LFG Parsing Coverage

Özlem Çetinoğlu **Sina Zarrieß** **Jonas Kuhn**
IMS, University of Stuttgart
{ozlem,zarriesa,jonas}@ims.uni-stuttgart.de

1 Introduction

Over the last two decades, the LFG community has witnessed the development of wide-coverage, deep, hand-crafted grammars for several languages (Butt et al. 2002). These grammars can typically parse over 90% of corpus data, yet cannot produce a full-fledged solution for each sentence. It is known that with a deep parsing approach 100% coverage is impossible due to idiosyncrasies, rare constructions, ungrammatical material, spelling errors in real language use.

We propose a scheme of sentence simplification and reparsing for sentences on which the original grammar fails. The guiding idea for the ultimate application of the scheme is the following: a state-of-the-art statistical dependency parser is run on any missed out sentence. Its coverage is 100%, and although the labelled dependency tree analysis that it produces will not be perfect, we can use it as a fairly reliable indication of "non-core" parts of the sentence: appositions, relative clauses, etc. We generate modified versions of the input string in which these parts are deleted and try to reparse them with the original LFG grammar. By using a conservative scheme of deletions, we try to ensure that grammaticality and the core argument structure of the sentences are preserved. Depending on the context of application, the resulting simplified f-structure can be used directly, or a synthesized analysis for the original string can be constructed (the latter option is of particular interest when using the LFG grammar to create training data for statistical generation, based on treebank data).

The full XLE parses of TIGER sentences are the basis for automatically extracted gold TIGER f-structures. So-called 'TIGER-compatible f-structures' are used in training the XLE parse disambiguation (Forst 2007) and generation ranking (Cahill et al. 2007). They are a natural application area for simplified f-structures. By matching them against simplified Tiger trees, we can produce more gold f-structures and increase the size of training data.

The focus of the present paper is as follows: we propose a set of simplification rules for the German ParGram grammar and test its effect on coverage relative to the standard TIGER treebank (Brants et al. 2002). To assess the effectiveness of the simplification rules, results on gold-standard dependency trees are most informative; in the full paper, we also report results on the type of trees an automatic dependency parser would produce (using a cross-validation technique).

2 Motivation

XLE deals with robustness with fragmenting and skimming (Riezler et al. 2002). However such mechanisms often lead to chunk and/or partial analyses. But actually, when an LFG grammar fails to produce a full parse for a sentence, it does not necessarily mean the best possible deep syntactic representation to achieve are fragments. (1) exemplifies a TIGER sentence, where the German ParGram Grammar gives only fragmented analyses. Removing only the word 'letztmals' makes it fully parsable again. All parts of the sentence but one are kept intact; the remaining part could still be highly beneficial for an application that uses the parser output.

(1) *Besitzer älterer Häuser können dieses Recht ebenfalls für*
Owners older houses can this right also for
*sich in Anspruch nehmen - **letztmals** für 1998 .*
themselves claim - **for the last time** in 1998 .
'Owners of older houses can also claim this right for themselves - for the last time in 1998.'

Sometimes the part of a sentence that causes a fragmented parse could be a phrase rather than a word. For instance, when the genitive adjunct 'des japanischen Außenministerium' is removed from (2), the simplified sentence is parsable by the German ParGram grammar. The missing 's' at the end of 'Außenministerium' makes the sentence ungrammatical and causes the parse failure in the original sentence.

(2) *Ein Sprecher **des japanischen Außenministerium** verkündete*
A speaker of the Japanese foreign ministry proclaimed
daraufhin , man werde Jelzins Aussage " vorsichtig
then , one would Jelzin's statement " carefully
analysieren " , bevor man sie kommentiere , aber :
analyze " , before one it comment , but :
'A speaker (of the Japanese foreign ministry) then proclaimed that Jelzin's statement would be " carefully analyzed " , before commenting on it , but :

This genitive adjunct can easily be found by using a dependency tree. Figure 1 depicts the dependency representation of the first six tokens of (2) and lists the deletable subtrees. Genitive adjuncts are one of the deletable labels as their removal would not harm the grammaticality. When the subtree labelled AG is deleted from the original tree, its yield is deleted from the original sentence.

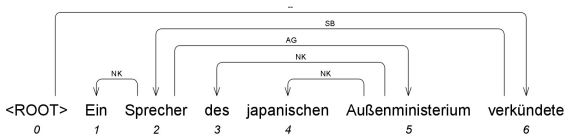

Figure 1: The dependency tree of the partial phrase *'Ein Sprecher des japanische Außenministerium verkündete'*. The subtree labelled with genitive adjunct (AG) is deletable. Subject (SB) is not deletable. Noun Kernel (NK) is deletable when the dependent is an adjective (*japanische*), not deletable when it is an article (*ein, des*).

3 Related Work

Rohrer and Forst (2006) showed that implementing additional rules for linguistic phenomena that cause non-full analyses improves the parsing coverage for the German Grammar. They also make use of skimming and fragmenting.

A completely different perspective for handling coverage problems could be to generate f-structures by annotating statistical phrase-structure parser output with f-structure constraints and by solving those constraints (Cahill et al. 2004). After all, the statistical parsers they are based on are robust. Although the approach is applied to several languages, only the English system's output approximates to XLE f-structure duplicates, which also cannot reach 100% coverage (Hautli et al. 2010)

Riezler et. al (2003) carried out sentence simplification by converting parsed f-structures to reduced ones with transfer rules. They then disambiguate and generate from reduced f-structures to obtain shorter sentences. Dependency based sentence simplification is a popular method, often used to extract the important information out of the data in applications such as semantic role labelling (Vickrey and Koller 2008), summarisation (Vanderwende et al. 2006), spoken language understanding (Tür et al. 2011) Some systems consider grammaticality in simplified candidates, whereas others do not.

4 Experiments

In our experiments, we parse the TIGER Treebank (Brants et al. 2002) 2.1 by using a version of the German ParGram grammar (Rohrer and Forst 2006). We leave out sentences from 8000 to 10000 as test and development sets following the TiGerDB split (Forst et al. 2004) and parse the rest.

For our simplification process we manually define a set of deletable dependency subtrees that would not harm the grammaticality of a sentence and preserve the core argument structure, such as modifiers, appositions, discourse markers.

As our initial experiment we delete only one deletable subtree each time. The number of candidates generated for each sentence varies depending on the number of deletable dependencies it contains. On average there are 5.6 candidates per sentence. In the second experiment, we generate all possible subtrees by deleting all combinations of deletable dependencies. As expected, the number of candidates grows very high for longer sentences, the highest number is 608,255. On average there are 924 candidates per sentence. For sentences with more than 10 candidates, we take the 10 shortest sentences to parse by XLE. The average of candidates per sentence drops down to 9 in this way.

Table 1 gives the overview of coverage statistics. 80.66% of the TIGER training set has full XLE parses[1]. The remaining 9373 sentences constitute the set to be simplified. When only one subtree is deleted, 3367 sentences have at least one simplified form with a full parse. When all possible candidates are created and 10 shortest are chosen, the number of sentences with at least one simplified full parse increases to 4607. Note that the upper limit of simplified sentences with a full parse is 8462 (90.28%) because 911 sentences are not simplifiable at all. We observe one of the major reasons of no full parses among simplified candidates is punctuation. It stems mainly from the representation of the punctuation itself. The TIGER Treebank does not necessarily attach all punctuation to a relevant node. The conversion tool aims to correctly identify the head node. Incorrect punctuation attachements are propagated through simplification iterations. Our simplification script handles general cases well but still needs improvement for more complex cases.

System	sent.	full parses
TIGER Training	48471	39098 (80.66%)
1 subtree shorter	9373	3367 (35.92%)
10 shortest	9373	4607 (49.83%)

Table 1: Full parse statistics when the original training sentences are used, only one subtree is deleted in simplification, and 10 shortest candidates are parsed among all possible simplifications.

[1] An additional application of sentence simplification which we leave aside here is to simplify fully parsed sentences that TIGER-compatibles f-structures cannot be extracted from. This could provide extra training material for parse disambiguation and generation ranking.

References

Brants, S., S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The tiger treebank.

Butt, M., T. H. King, H. Masuichi, and C. Rohrer. 2002. The parallel grammar project. In N. J. Carroll and R. Sutcliffe (Eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation*, 1–7. COLING02.

Cahill, A., M. Burke, R. O'Donovan, J. Van Genabith, and A. Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage pcfg-based lfg approximations. In *(ACL'04)*.

Cahill, A., M. Forst, and C. Rohrer. 2007. Stochastic realisation ranking for a free word order language. In *ENLG '07*.

Forst, M. 2007. *Disambiguation for a Linguistically Precise German Parser*. PhD thesis, University of Stuttgart.

Forst, M., N. Bertomeu, B. Crysmann, F. Fouvry, S. Hansen-Schirra, and V. Kordoni. 2004. Towards a dependency-based gold standard for German parsers – The TiGer Dependency Bank. In *LINC '04*.

Hautli, A., O. Çetinoğlu, and J. van Genabith. 2010. Closing the gap between stochastic and hand-crafted lfg grammars. In *Proceedings of the LFG Conference*.

Riezler, S., T. H. King, R. Crouch, and A. Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lfg. In *HLT-NAACL*.

Riezler, S., T. H. King, R. M. Kaplan, R. Crouch, J. T. Maxwell, and I. M. Johnson. 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *ACL 2002*.

Rohrer, C., and M. Forst. 2006. Improving coverage and parsing quality of a large-scale lfg for german. In *LREC 2006*.

Tür, G., D. Hakkani-Tür, L. Heck, and S. Parthasarathy. 2011. Sentence Simplification for Spoken Language Understanding. In *IEEE ICASSP*.

Vanderwende, L., H. Suzuki, and C. Brockett. 2006. Microsoft Research at DUC 2006: Task-Focused Summarization with Sentence Simplification and Lexical Expansion. In *Document Understanding Workshop, HLT-NAACL 2006*.

Vickrey, D., and D. Koller. 2008. Sentence simplification for semantic role labeling. In *ACL-08: HLT*.