Improving data-driven dependency parsing using large-scale LFG grammars

Lilja Øvrelid, Jonas Kuhn and Kathrin Spreyer
Department of Linguistics
University of Potsdam
{lilja,kuhn,spreyer}@ling.uni-potsdam.de

Abstract

This paper presents experiments which combine a grammar-driven and a data-driven parser. We show how the conversion of LFG output to dependency representation allows for a technique of parser stacking, whereby the output of the grammar-driven parser supplies features for a data-driven dependency parser. We evaluate on English and German and show significant improvements stemming from the proposed dependency structure as well as various other, deep linguistic features derived from the respective grammars.

1 Introduction

The divide between grammar-driven and data-driven approaches to parsing has become less pronounced in recent years due to extensive work on robustness and efficiency for the grammar-driven approaches (Riezler et al., 2002; Cahill et al., 2008b). The linguistic generalizations captured in such knowledge-based resources are thus increasingly available for use in practical applications.

The NLP-community has in recent years witnessed a surge of interest in dependency-based approaches to syntactic parsing, spurred by the CoNLL shared tasks of dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007). Nivre and McDonald (2008) show how two different approaches to dependency parsing, the graph-based and transition-based approaches, may be combined and subsequently learn to complement each other to achieve improved parse results for a range of different languages.

In this paper, we show how a data-driven dependency parser may straightforwardly be modified to learn directly from a grammar-driven parser. We evaluate on English and German and show significant improvements for both languages. Like Nivre and McDonald (2008), we supply a data-driven dependency parser with features from a different parser to guide parsing. The additional parser employed in this work, is not however, a data-driven parser trained on the same data set, but a grammar-driven parser outputing a deep LFG analysis. We furthermore show how a range of other features – morphological, structural and semantic – from the grammar-driven analysis may be employed during data-driven parsing and lead to significant improvements.

2 Grammar-driven LFG-parsing

The XLE system (Crouch et al., 2007) performs unification-based parsing using hand-crafted LFG grammars. It processes raw text and assigns to it both a phrase-structural (‘c-structure’) and a feature structural, functional (‘f-structure’).

In the work described in this paper, we employ the XLE platform using the grammars available for English and German from the ParGram project (Butt et al., 2002). In order to increase the coverage of the grammars, we employ the robustness techniques of fragment parsing and ‘skimming’ available in XLE (Riezler et al., 2002).

3 Dependency conversion and feature extraction

In extracting information from the output of the deep grammars we wish to capture as much of the precise, linguistic generalizations embodied in the grammars as possible, whilst keeping with the requirements posed by the dependency parser. The process is illustrated in Figure 1.

3.1 Data

The English data set consists of the Wall Street Journal sections 2-24 of the Penn treebank (Marcus et al., 1993), converted to dependency format. The treebank data used for German is the Tiger

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 37–40,
Suntec, Singapore, 4 August 2009.

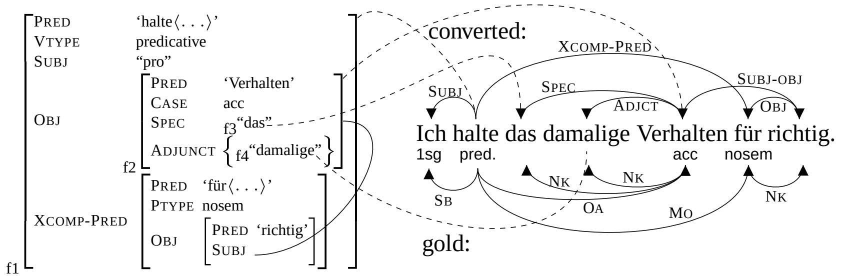

Figure 1: Treebank enrichment with LFG output; German example: *I consider the past behaviour correct.*

treebank (Brants et al., 2004), where we employ the version released with the CoNLL-X shared task on dependency parsing (Buchholz and Marsi, 2006).

3.2 LFG to dependency structure

We start out by converting the XLE output to a dependency representation. This is quite straightforward since the f-structures produced by LFG parsers can be interpreted as dependency structures. The conversion is performed by a set of rewrite rules which are executed by XLE's built-in extraction engine. We employ two strategies for the extraction of dependency structures from output containing multiple heads. We attach the dependent to the closest head and, i) label it with the corresponding label (Single), ii) label it with the complex label corresponding to the concatenation of the labels from the multiple head attachments (Complex). The converted dependency analysis in Figure 1 shows the f-structure and the corresponding converted dependency output of a German example sentence, where a raised object *Verhalten* receives the complex SUBJ-OBJ label. Following the XLE-parsing of the treebanks and the ensuing dependency conversion, we have a grammar-based analysis for 95.2% of the English sentence, 45238 sentences altogether, and 96.5% of the German sentences, 38189 sentences altogether.

3.3 Deep linguistic features

The LFG grammars capture linguistic generalizations which may not be reduced to a dependency representation. For instance, the grammars contain information on morphosyntactic properties such as case, gender and tense, as well as more semantic properties detailing various types of adverbials, specifying semantic conceptual categories such as human, time and location etc., see Figure 1. Table 1 presents the features extracted for use during parsing from the German and English XLE-parses.

4 Data-driven dependency parsing

MaltParser (Nivre et al., 2006a) is a language-independent system for data-driven dependency parsing which is freely available.[1] MaltParser is based on a deterministic parsing strategy in combination with treebank-induced classifiers for predicting parse transitions. MaltParser constructs parsing as a set of transitions between parse configurations. A parse configuration is a triple $\langle S, I, G\rangle$, where S represents the parse stack, I is the queue of remaining input tokens, and G represents the dependency graph defined thus far.

The feature model in MaltParser defines the relevant attributes of tokens in a parse configuration. Parse configurations are represented by a set of features, which focus on attributes of the *top* of the stack, the *next* input token and neighboring tokens in the stack, input queue and dependency graph under construction. Table 2 shows an example of a feature model.[2]

For the training of baseline parsers we employ feature models which make use of the word form (FORM), part-of-speech (POS) and the dependency relation (DEP) of a given token, exemplified in Table 2. For the baseline parsers and all subsequent parsers we employ the arg-eager algorithm in combination with SVM learners with a polynomial kernel.[3]

[1] http://maltparser.org

[2] Note that the feature model in Table 2 is an example feature model and not the actual model employed in the parse experiments. The details or references for the English and German models are provided below.

[3] For training of the baseline parsers we also employ some language-specific settings. For English we use learner and parser settings, as well as feature model from the English pretrained MaltParser-model available from http://maltparser.org. For German, we use the learner and parser settings from the parser employed in the CoNLL-X

POS	XFeats
Verb	CLAUSETYPE, GOVPREP, MOOD, PASSIVE, PERF, TENSE, VTYPE
Noun	CASE, COMMON, GOVPREP, LOCATIONTYPE, NUM, NTYPE, PERS, PROPERTYPE
Pronoun	CASE, GOVPREP, NUM, NTYPE, PERS
Prep	PSEM, PTYPE
Conj	COORD, COORD-FORM, COORD-LEVEL
Adv	ADJUNCTTYPE, ADVTYPE
Adj	ATYPE, DEGREE
English	DEVERBAL, PROG, SUBCAT, GENDSEM, HUMAN, TIME
German	AUXSELECT, AUXFLIP, COHERENT, FUT, DEF, GEND, GENITIVE, COUNT

Table 1: Features from XLE output, common for both languages and language-speciffic

	FORM	POS	DEP	XFEATS	XDEP
S:*top*	+	+	+	+	+
I:*next*	+	+		+	+
I:*next*-1	+			+	
G:head of *top*	+			+	
G:leftmost dependent of *top*			+	+	
InputArc(XHEAD)					

Table 2: Example feature model; S: stack, I: input, G: graph; $\pm n$ = n positions to the left($-$) or right ($+$).

5 Parser stacking

The procedure to enable the data-driven parser to learn from the grammar-driven parser is quite simple. We parse a treebank with the XLE platform. We then convert the LFG output to dependency structures, so that we have two parallel versions of the treebank – one gold standard and one with LFG-annotation. We extend the gold standard treebank with additional information from the corresponding LFG analysis, as illustrated by Figure 1 and train the data-driven dependency parser on the enhanced data set.

We extend the feature model of the baseline parsers in the same way as Nivre and McDonald (2008). The example feature model in Table 2 shows how we add the proposed dependency relation (XDEP) *top* and *next* as features for the parser. We furthermore add a feature which looks at whether there is an arc between these two tokens in the dependency structure (InputArc(XHEAD)), with three possible values: Left, Right, None. In order to incorporate further information supplied by the LFG grammars we extend the feature models with an additional, static attribute, XFEATS. This is employed for the range of deep linguistic features, detailed in section 3.3 above.

5.1 Experimental setup

All parse experiments are performed using 10-fold cross-validation for training and testing. Overall parsing accuracy will be reported using the standard metrics of labeled attachment score (LAS) and unlabeled attachment score (UAS).Statistical significance is checked using Dan Bikel's randomized parsing evaluation comparator.[4]

6 Results

We experiment with the addition of two types of features: i) the dependency structure proposed by XLE for a given sentence ii) other morphosyntactic, structural or lexical semantic features provided by the XLE grammar. The results are presented in Table 3.

For English, we find that the addition of proposed dependency structure from the grammar-driven parser causes a small, but significant improvement of results (p<.0001). In terms of labeled accuracy the results improve with 0.15 percentage points, from 89.64 to 89.79. The introduction of complex dependency labels to account for multiple heads in the LFG output causes a smaller improvement of results than the single labeling scheme. The corresponding results for German are presented in Table 3. We find that the addition of grammar-driven dependency structures with single labels (Single) improves the parse results significantly (p<.0001), both in terms of unlabeled and labeled accuracy. For labeled accuracy we observe an improvement of 1.45 percentage points, from 85.97 to 87.42. For the German data, we find that the addition of dependency structure with complex labels (Complex) gives a further small, but significant (p<.03) improvement over the experiment with single labels.

The results following the addition of the grammar-extracted features in Table 1 (Feats) are presented in Table 3.[5] We observe significant improvements of overall parse results for both languages (p<.0001).

shared task (Nivre et al., 2006b). For both languages, we employ so-called "relaxed" root handling.

[4] http://www.cis.upenn.edu/∼dbikel/software.html

[5] We experimented with several feature models for the inclusion of the additional information, however, found no significant differences when performing a forward feature selection. The simple feature model simply adds the XFEATS of the *top* and *next* tokens of the parse configuration.

	English		German	
	UAS	LAS	UAS	LAS
Baseline	92.48	89.64	88.68	85.97
Single	**92.61**	**89.79**	89.72	87.42
Complex	92.58	89.74	89.76	87.46
Feats	92.55	89.77	89.63	87.30
Single+Feats	92.52	89.69	90.01	87.77
Complex+Feats	92.53	89.70	**90.02**	**87.78**

Table 3: Overall results in experiments expressed as unlabeled and labeled attachment scores.

We also investigated combinations of the different sources of information – dependency structures and deep features. These results are presented in the final lines of Table 3. We find that for the English parser, the combination of the features do not cause a further improvement of results, compared to the individual experiments. The combined experiments (Single+Feats, Complex+Feats) for German, on the other hand, differ significantly from the baseline experiment, as well as the individual experiments (Single,Complex,Feats) reported above ($p<.0001$). By combination of the grammar-derived features we improve on the baseline by 1.81 percentage points.

A comparison with the German results obtained using MaltParser with graph-based dependency structures supplied by MSTParser (Nivre and McDonald, 2008) shows that our results using a grammar-driven parser largely corroborate the tendencies observed there. Our best results for German, combining dependency structures and additional features, slightly improve on those reported for MaltParser (by 0.11 percentage points).[6]

7 Conclusions and future work

This paper has presented experiments in the combination of a grammar-driven LFG-parser and a data-driven dependency parser. We have shown how the use of converted dependency structures in the training of a data-driven dependency parser, MaltParser, causes significant improvements in overall parse results for English and German. We have furthermore presented a set of additional, deep features which may straightforwardly be extracted from the grammar-based output and cause individual improvements for both languages and a combined effect for German.

In terms of future work, a more extensive error analysis will be performed to locate the precise benefits of the parser combination. We will also investigate the application of the method directly to raw text and application to a task which may benefit specifically from the combined analyses, such as semantic role labeling or semantic verb classification.

It has recently been shown that automatically acquired LFG grammars may actually outperform hand-crafted grammars in parsing (Cahill et al., 2008a). These results add further to the relevance of the results shown in this paper, bypassing the bottleneck of grammar hand-crafting as a prerequisite for the applicability of our results.

[6] English was not among the languages investigated in-Nivre and McDonald (2008).

References

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther Knig, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. Tiger: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2:597–620.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X).*

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation*.

Aoife Cahill, Michael Burke, Ruth O'Donovan, Stefan Riezler, Josef van Genabith, and Andy Way. 2008a. Wide-coverage deep statistical parsing using automatic dependency structure annotation. *Computational Linguistics*.

Aoife Cahill, John T. Maxwell, Paul Meurer, Christian Rohrer, and Victoria Rosen. 2008b. Speeding up LFG parsing using c-structure pruning. In *Proceedings of the Workshop on Grammar Engineering Across Frameworks*.

D. Crouch, M. Dalrymple, R. Kaplan, T. King, J. Maxwell, and P. Newman, 2007. *XLE Documentation*. http://www2.parc.com/isl/.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus for English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-HLT 2008.*

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006a. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC.*

Joakim Nivre, Jens Nilsson, Johan Hall, Gülşen Eryiğit, and Svetoslav Marinov. 2006b. Labeled pseudo-projective dependency parsing with Support Vector Machines. In *Proceedings of CoNLL.*

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007,* pages 915–932.

Stefan Riezler, Tracy King, Ronald Kaplan, Richard Crouch, John T. Maxwell, and Mark Johnson. 2002. Parsing the Wall Street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of ACL.*