Cross-framework parser stacking for data-driven dependency parsing

Lilja Øvrelid, Jonas Kuhn and Kathrin Spreyer

Department of Linguistics, University of Potsdam
Karl-Liebknecht-Str 24/25, 14476 Potsdam
{ovrelid,kuhn,spreyer}@uni-potsdam.de

ABSTRACT. *In this article, we present and evaluate an approach to the combination of a grammar-driven and a data-driven parser which exploits machine learning for the acquisition of syntactic analyses guided by both parsers. We show how 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 in overall parse results stemming from the proposed dependency structure as well as other linguistic features derived from the grammars. Finally, we perform an application-oriented evaluation and explore the use of the stacked parsers as the basis for the projection of dependency annotation to a new language.*

RÉSUMÉ. *Dans cet article, nous présentons et évaluons une approche permettant de combiner un analyseur fondé sur une grammaire et un analyseur fondé sur des données, en utilisant des méthodes d'apprentissage automatique pour produire des analyses syntaxiques guidées par les deux analyseurs. Nous montrons comment la conversion de la sortie d'un analyseur LFG en une représentation en dépendances permet d'utiliser une technique d'empilement d'analyseurs ("parser stacking"), dans laquelle la sortie de l'analyseur fondé sur une grammaire fournit des caractéristiques utilisables par un analyseur fondé sur les données. Nous évaluons notre approche sur l'anglais et l'allemand, et montrons des améliorations significatives pour les résultats d'analyses syntaxiques complètes qui découlent de l'analyse en dépendances ainsi que des caractéristiques provenant de grammaires. Enfin, nous procédons à une évaluation dédiée à une application, et explorons l'utilisation de cet empilement d'analyseurs comme point de départ pour l'annotation en dépendances d'une nouvelle langue.*

KEYWORDS: *data-driven dependency parsing, Lexical Functional Grammar (LFG), parser combination, stacking, deep linguistic features*

MOTS-CLÉS : *analyse syntaxique en dépendances fondé sur des données, Grammaires Lexicales Fonctionnelles (LFG), combinaison d'analyseurs, caractéristiques linguistiques profondes*

TAL. Volume 50 – n° 3/2009, pages 109 à 138

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; Hockenmaier and Steedman, 2002; Miyao and Tsujii, 2005; Clark and Curran, 2007; Zhang *et al.*, 2007; Cahill *et al.*, 2008a; Cahill *et al.*, 2008b). Hybrid techniques which combine hand-crafted linguistic knowledge with statistical models for parse disambiguation characterize most large-scale grammar-driven parsing systems. 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 data-driven dependency parsing, the graph-based and transition-based approaches, may be combined and subsequently learn to complement each other to achieve improved parsing results for a range of different languages.

Although there has been some work towards incorporating more linguistic knowledge in data-driven parsing by means of feature design or representational choices (Klein and Manning, 2003; Bod, 1998; Øvrelid and Nivre, 2007), few studies investigate a setting where a data-driven parser may learn directly from a grammar-driven one.[1] In this paper, we show how a data-driven dependency parser may straightforwardly be modified to learn from a grammar-driven parser, hence combining the strengths of the two approaches to syntactic parsing. We investigate an approach which relies only on a general mapping from grammar output to dependency graphs. We evaluate on English and German and show significant improvements for both languages in terms of overall parsing results, stemming both from a dependency structure representation proposed by the grammar-driven parser and a set of additional features extracted from the respective grammars. A detailed feature and error analysis provides further insight into the precise effect of the linguistic, grammar-derived information in parsing and the differences between the two languages. We furthermore investigate the importance of parser quality in the parser stacking setting. Experiments with automatically assigned part-of-speech tags set the scene for an application-realistic setting and we show how very similar and significant improvements may be obtained in the application of the parser combination to raw text. Finally, we go on to explore a realistic example of the use of the stacked parser in a more complex application scenario, which among other things involves out-of-domain application of the components: we apply the parsers as the basis for the projection of dependency annotation to a new language and show significant improvements of results for this task compared to baseline parsers.

1. See Zhang and Wang (2009), however, for a similar study which exploits an English HPSG grammar during parsing and applies the resulting system to an out-of-domain parsing task.

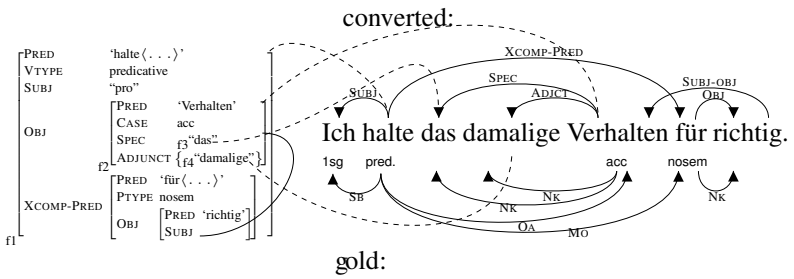

Figure 1. *Treebank enrichment with LFG output for German example sentence* Ich halte das damalige Verhalten für richtig *'I consider the past behavior (to be) correct'*

The paper is structured as follows. Section 2 briefly introduces grammar-driven LFG-parsing, and Section 3 describes the conversion to dependency structures and the feature extraction. In Section 4 we introduce MaltParser, the data-driven dependency parser employed in the experiments, and Section 5 goes on to describe the parser stacking experiments in more detail. Section 6 presents the results from the experiments, and Section 7 provides an in-depth error analysis of these results. The effect of using automatically assigned part-of-speech tags is examined in a set of experiments detailed in Section 8. In Section 9 we present experiments with the stacked parsers in the task of annotation projection. Finally, Section 10 concludes and discusses the generality of the approach and some ideas for future work.

2. Grammar-driven LFG-parsing

The ParGram project (Butt *et al.*, 2002) has resulted in wide coverage grammars for a number of languages. These grammars have been written collaboratively within the linguistic framework of Lexical Functional Grammar (LFG) and employ a common set of grammatical features. In the work described in this paper, we use the English and German grammars from the ParGram project. The XLE system (Crouch *et al.*, 2007) performs unification-based parsing using these hand-crafted grammars, assigning a LFG analysis. It processes raw text and assigns to it both a phrase-structural ('c-structure') and a feature-structural, functional ('f-structure') representation which encodes predicate-argument structure. The set of grammatically possible parsing analyses for the input string is represented in a packed format and then undergoes a statistical disambiguation step: a log-linear model that has been trained on a standard treebank (Riezler *et al.*, 2002) is used to single out the most probable analysis.

In a dependency-based evaluation[2] on a subset of Penn treebank sentences, the English grammar was earlier reported to achieve a 77.6% F-score (Kaplan *et al.*, 2004), whereas the German grammar achieves a 72.59% F-score (Forst *et al.*, 2004) on the Tiger treebank. 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, while keeping to the requirements of the dependency parser. This means that the deep analysis must be reduced to a token-based representation. The process is illustrated in Figure 1 and details of the conversion process are provided in Section 3.2 below.

3.1. *Data*

We make use of standard data sets for the two languages. The English data set consists of the *Wall Street Journal* sections 2-24 of the Penn treebank for English (Marcus *et al.*, 1993), converted to dependency format (Johansson and Nugues, 2007). The treebank data used for German is the Tiger treebank for German (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*

The parser stacking relies on a conversion of LFG output to dependency graphs, so we start out by extracting a dependency representation from the XLE output. The extraction is performed by a set of rewrite rules which are executed by XLE's built-in extraction engine. The mapping between input tokens and f-structures is readily available in the LFG analysis via the ϕ-projection. In Figure 1, the dashed lines illustrate the mapping for the object NP *das damalige Verhalten* in example (3) below. The head of the NP (*Verhalten*) maps to the f-structure *f2*, which has *f3* (identified with *das*) as its specifier (SPEC) and *f4* (*damalige*) as an adjunct (ADJCT). These dependencies are straightforwardly adopted in the extracted dependency representation as shown. In Figure 1, the gold standard treebank analysis is shown below the sentence

2. F-structures are reduced to dependency-triples and evaluation is performed using the standard measures of precision and recall. A subset of the triples, i.e. the dependencies ending in a 'pred'-value, provides the basis for the evaluation scores presented here.

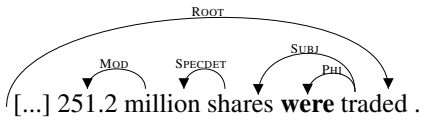

Figure 2. *Example of generic* PHI*-arc between co-heads of the verbal complex* were traded

('gold') and the resulting converted dependency analysis ('converted') is shown above the sentence.

The mapping between input tokens and f-structures represented via the ϕ-projection is not necessarily injective. This means that two tokens may be mapped to the same f-structure, e.g., the auxiliary and the main verb in the English present perfect construction. While this lack of isomorphy allows for highly modular representations that clearly distinguish between surface-oriented properties (c-structure) and abstract syntactic and semantic dimensions (f-structure) including the additional linguistic features described in Section 3.3 below, it is not directly compatible with the word-based representations assumed in dependency grammar. We reconcile this discrepancy by postulating generic arcs (labeled PHI) between such co-heads, such that the token that introduces the predicate to the f-structure is the head of all its co-heads in the dependency structure. This is illustrated in Figure 2, where the passive auxiliary *were* becomes the PHI-dependent of *traded* in the derived dependency structure, because they are mapped to the same f-structure in the XLE output.

Most dependency parsers require that sentences be represented as trees where each token is a node and dependents have exactly one head. LFG parsers, on the other hand, compute structures which are directed acyclic graphs, where a dependent may have more than one head. Within LFG, so-called 'structure sharing' is employed to account for phenomena such as raising and control. In the conversion process, we attach dependents with multiple heads to their closest head and supply them with the corresponding label. In an alternative version, we also represent the additional attachment as a set of complex dependency labels listing the functional paths. For instance, in the English example in (1), a so-called 'tough' construction (Rosenbaum, 1967; Postal and Ross, 1971) where the subject of the matrix verb is also an object of the infinitival clause, or in the subject-to-object raising construction in (2), the boldfaced argument receives the complex label SUBJ-OBJ. In German, we find similar analyses of object predicative constructions, for instance, as in (3). The converted dependency analysis in Figure 1 shows the f-structure and the corresponding converted dependency output of example (3), where *Verhalten* receives the complex SUBJ-OBJ label.

(1) *the* **anthers** *in these plants are difficult to clip*

(2) *allowed the* **company** *to begin marketing a new lens*

(3) *Ich halte das damalige* **Verhalten** *für richtig*
I hold the past behavior for right
'I consider the past behavior (to be) correct'

Since the LFG grammars and the treebank annotation make somewhat different assumptions with respect to tokenization, a stage of post-processing of the extracted data is necessary in order to make the annotations truly parallel. In particular, the treatment of multiword units, hyphenated expressions, and sentence-internal punctuation, such as initials and genitive *'s* in English, is mapped to match the treebank annotation in a subsequent post-processing stage. This stage improves on the level of token-wise parallelism between the two versions of the treebank – the gold standard and the XLE-parsed, converted version – and has a quite dramatic effect, illustrating the importance of this stage. The level of mismatch between the two versions, i.e. the number of tokens that are not mapped to a grammar-derived analysis, is reduced following post-processing by 85.8% for English and 66.8% for German. The dramatic effect of this process is mostly due to propagation of mismatch within a sentence. For instance, the XLE grammar for English and German treats multiword units like *New York* as one word form, whereas the treebank annotations split these into two forms. On the other hand, the grammar treats hyphenated expressions in English, such as *needle-like*, as one form, whereas the treebank does not. In terms of the effect of post-processing on the dependency analysis of the grammar, we adopt the following strategy: we duplicate the analysis of composite forms which are split, like *New York*, and, we assign to concatenated forms the analysis of its last element, as in the hyphenated *needle-like*.

3.3. *Additional linguistic features*

The LFG grammars contain linguistic generalizations which may not be reduced to a dependency structure. 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, distinguishing count and mass nouns, specifying semantic conceptual categories such as human, time and location, etc.

Table 1 presents the features extracted for use during parsing from the German and English XLE-parses, organized by the part-of-speech of the dependent. The final two rows of Table 1 provide features which are language-specific for the two languages (English and German), whereas the rest are shared features of both grammars. We also provide the possible values for the features, most of which are atomic or binary-valued. A few features (GOVPREP, COORDFORM) take the word form (Form) of another token in the syntactic context as feature value. Due to the fact that LFG is a unification-based formalism, some of the features will not be strictly local to a specific token, but stem from a head or dependent (or mother/sister, to be precise).

For instance, the German definiteness feature (DEF) of a common noun comes from the determiner. The features also bear witness to the fact that XLE makes use of a phrase-structural representation for parsing (c-structure). For instance, it provides a COORD-LEVEL feature which takes as values phrase categories such as NP and VP, detailing the phrasal level of coordination.

Quite a few of the features detailed in Table 1 are available in the original treebank annotation. Clearly, treebanks differ quite a lot in the amount of information expressed by the annotation. The Penn Treebank contains only information on part-of-speech and syntactic structure, whereas the Tiger treebank in addition distinguishes the morphosyntactic categories case, number, gender, person, tense and mood. The level of linguistic analysis expressed by the features varies ranging from morphosyntactic information, which is fairly straightforward to obtain, to deeper features detailing structural aspects (GOVPREP, COORDFORM), subcategorization information (VTYPE, SUBCAT) or various semantic properties which reflect more fine-grained linguistic analyses of phenomena such as nominal semantics (COMMON, PROPERTYPE etc.), referentiality (NTYPE, DEIXIS, GENDSEM), adverbial semantics (ADJUNCTTYPE, ADVTYPE), nominalization (DEVERBAL), etc. Whether these distinctions prove to be helpful in the task of syntactic parsing and which of them contribute the most is an empirical question which we will investigate experimentally below.

3.4. *Coverage*

Hand-crafted grammars are often accused of low coverage on running texts. However, a lot of effort has in recent years been put into increasing their robustness. As mentioned earlier, we employ the robustness techniques available with XLE, allowing for fragmented parses as well as using the technique of 'skimming', which sets a bounded amount of work performed per subtree and skims the remaining constituents when too much time has passed (Riezler *et al.*, 2002).

Following the XLE-parsing of the treebanks and the ensuing dependency conversion, we have sentence coverage of 95.4% for the English grammar and 97.3% for the German grammar. In order to align the two versions of the treebank we then perform tokenization fixes for the respective languages, as described in Section 3.2 above. Following this stage of post-processing, we find that 92.3% of the English tokens and 95.2% of the German tokens may be mapped to the corresponding token in the treebank. In terms of sentence coverage, we end up with a grammar-based analysis for 95.2% of the English sentences, 45238 sentences altogether, and 96.5% of the German sentences, 38189 sentences altogether.

POS		Possible values of features
Verb	CLAUSETYPE	conditional, declarative, imperative, interrogative
	MOOD	imperative, indicative, subjunctive
	TENSE	future, past, present
	VTYPE	copula, main, modal, predicative, raising
	PASSIVE(+/−), PERF(+/−), GOVPREP(Form)	
Noun/Pro	CASE	genitive, nominative, oblique/dative, accusative
	COMMON	count, gerund, measure, partitive
	LOCATIONTYPE	city, country
	NUM	plural, singular
	NTYPE	common, proper/demon, expl, free, int, locative, null, pers, poss, quant, recip, refl, rel
	PERS	1st, 2nd, 3rd
	PROPERTYPE	company, location, name, organization, title
	GOVPREP(Form)	
Preposition	PSEM	direction, location, manner
	PTYPE	nosem, sem
Coordination	COORDLEVEL	ADVP, AP, CP, NP, Cbar, PP, VP, etc.
	COORD(+/−), COORDFORM(Form)	
Adverbs	ADJUNCTTYPE	affix, conditional, degree, label, negation, parenth, rel
	ADVTYPE	focus, sadv, vpadv/dir, loc, manner, temp, unspec
Adjectives	ATYPE	attributive, predicative
English	DEIXIS	distal, proximal
	DEVERBAL	passive, progressive
	SUBCAT	V-SUBJ,V-SUBJ-OBJ,V-SUBJ-COMPthat, etc.
	GENDSEM	female, male, nonhuman
	TIME	clock-time, date, season, day, month, year
	PROG(+/−), HUMAN(+/−)	
German	AUXSELECT	haben, sein
	GEND	feminine, masculine, neuter
	PARTICIPLE	perfect, present
	AUXFLIP(+/−), COHERENT(+/−), COUNT(+/−), DEF(+/−), FUT(+/−), GENITIVE(+/−)	

Table 1. *Features from the XLE output with possible values, common for both languages and language-specific (English and German)*

4. Data-driven dependency parsing

MaltParser (Nivre *et al.*, 2006b; Nivre *et al.*, 2006a) is a language-independent system for data-driven dependency parsing which is freely available.[3] It is based on a deterministic parsing strategy in combination with treebank-induced classifiers for predicting parsing actions. MaltParser employs a rich feature representation of the parse history in order to guide parsing, and may easily be extended to take into account new features of the parse history.

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 – a list of

3. http://maltparser.org

	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 (*$+$*)*

tokens which are candidates for dependency arcs – I is the queue of remaining input tokens, and G represents the dependency graph under construction. The parse *guide* predicts the next parse action (transition), based on the current parse configuration. The guide is trained employing discriminative machine learning, which recasts the learning problem as a classification problem: given a parse configuration, predict the next transition.

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 which employs the word form (FORM), part of speech (POS), and dependency relation (DEP) of a given token.[4] The feature model is depicted as a matrix where rows denote tokens in the parser configuration, defined relative to the stack (S), input queue (I) and dependency graph (G), and columns denote attributes. Each cell containing a $+$ corresponds to a feature of the model. Examples of the features include part-of-speech for the top of the stack, lexical form for the next and previous (*next-1*) input tokens and the dependency relation of the rightmost sibling of the leftmost dependent of *top*.

5. Parser stacking

The procedure for enabling the data-driven parser to learn from the grammar-driven parser is quite simple. We parse a treebank employed for training the data-driven baseline parser 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 and train the data-driven dependency parser on the enhanced data set.

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

ID	FORM	POS	FEATS	HEAD	DEPREL	XHEAD	XDEP
1	At	IN	psem:_|ptype:sem	11	TMP	12	ADJUNCT
2	the	DT	_	3	NMOD	3	SPECDET
3	end	NN	num:sg|pers:3|case:obl|common:count	1	PMOD	1	OBJ
4	of	IN	ptype:sem	3	NMOD	3	ADJUNCT
5	the	DT	_	6	NMOD	6	SPECDET
6	day	NN	govPrep:of|case:obl|time:+|num:sg|pers:3	4	PMOD	4	OBJ
7	,	,	_	11	P	0	PUNC
8	251.2	CD	_	9	DEP	9	MOD
9	million	CD	_	10	NMOD	10	SPECDET
10	shares	NNS	num:pl|common:count|pers:3|case:nom	11	SBJ	12	SUBJ
11	were	VBD	_	0	ROOT	12	PHI
12	traded	VBN	tense:past|subcat:V-SUBJ-OBJ|passive:+	11	VC	0	ROOT
13	.	.	_	11	P	0	PUNC

Table 3. *Enhanced treebank version of the English example sentence* At the end of the day, 251.2 million shares were traded

Table 3 shows the enhanced treebank version of the English sentence in example (4). For each token, the treebank contains information on word form, in column 2 (FORM), part-of-speech tag, in column 3 (POS), as well as the head and dependency relation in rows 5-6 (HEAD, DEPREL) in Table 3. The added XLE information resides in the FEATS-column, column 4, and in the additional columns 7-8 (XHEAD, XDEP) in Table 3.[5]

(4) At the end of the day, 251.2 million shares were traded.

It is clear already from this example that there are some interesting differences between the two annotations. The treebank annotation makes the finite verb *were* (row 11 in Table 3) the head of the whole dependency tree by attachment to the artificial root node 0 with dependency label ROOT, whereas the LFG analysis makes the lexical verb *traded* the root. It follows that arguments, such as the subject node *shares* in (4), are attached to different nodes in the two annotation schemes.

There is furthermore the fact that the treebank annotation has been manually checked, whereas the LFG output has not. The latter is thus bound to contain errors, which will certainly add noise to the training data provided for the data-driven parser. That being said, we may also expect that the errors made by the two parsers are qualitatively different due to the fundamental differences in the parser – the grammar-driven parser will typically suffer from missing rules or lexical entries, whereas the data-driven parser will be constrained by the types of structures found in the training data.

5. The CoNLL-format also contains columns for information about lemma and a more fine-grained part-of-speech tag. These columns are, however, empty in our data sets and were therefore omitted from Table 3.

Given the differences between the two annotations, it will be interesting to see whether these differences will turn out to complement each other and whether the data-driven parser will actually learn to generalize from the linguistic insights expressed in the grammar-driven system. The next sections are devoted to various aspects of this topic.

5.1. *Parser modifications*

In order for the data-driven parser to make use of the grammar-driven analyses both during learning and parsing, we make some modifications to the baseline feature models described in Section 4. We extend the feature model of the baseline parsers using the technique employed in Nivre and McDonald (2008). This allows us to add the predictions of another parser, or several other parsers, as features for the current parser. In this case we want to add the dependency substructure proposed by the grammar-driven parser as a feature for our data-driven parser. We thus need to be able to refer to the head (XHEAD) and dependency relation (XDEP) proposed by the grammar-driven system, for each token in a parse configuration. The example feature model in Table 2 shows how we add the proposed dependency relation (XDEP) for the token on top of the stack (*top*) and for the next input token (*next*) as features for the parser. We also 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 additional linguistic features, detailed in Section 3.3 above.

5.2. *Experimental setup*

For the training of baseline parsers we employ feature models which make use of the FORM, POS and DEP features exemplified in Table 2. For the baseline parsers and all subsequent parsers we employ the arc-eager algorithm (Nivre, 2003) in combination with SVM learners, using LIBSVM (Chang and Lin, 2001) with a polynomial kernel.[6] We employ the following language-specific settings:

English: Learner and parser settings, as well as a feature model from the English pretrained MaltParser-model.[7]

6. For both languages, we employ so-called "relaxed" root handling, which allows for root dependents to remain unattached during parsing and hence for the reduction of unattached tokens. This was found to improve results for the baseline parsers for both languages.

7. Available from http://maltparser.org

German: Learner and parser settings from the German parser employed in the CoNLL-X shared task (Nivre *et al.*, 2006b). We also employ the technique of pseudo-projective parsing described in Nilsson and Nivre (2005).

All parsing experiments are performed using ten-fold cross-validation for training and testing. This gives us as large as possible a sample of each language and more examples of less frequent constructions, e.g., control and raising constructions. Overall parsing accuracy will be reported using the standard metrics of *labeled attachment score* (LAS) and *unlabeled attachment score* (UAS). These report the percentage of tokens that are assigned the correct head *with* (labeled) or *without* (unlabeled) the correct dependency label.

Following the evaluation setup from the CoNLL shared tasks on dependency parsing, statistical significance is checked using Dan Bikel's randomized parsing evaluation comparator, and we report the average p-value over the ten cross-validations along with standard deviation (σ).[8] The experiments are performed using gold standard part-of-speech tags; however, there is nothing that prevents the same technique to be directly applied to raw text. As mentioned earlier, XLE comes with its own tokenizer and part-of-speech tagger. We will return to this point in Section 8.

6. Results

We perform a set of experiments investigating parser stacking for English and German, employing converted output from the grammar-driven system assigning a LFG analysis. We experiment with the addition of two types of features: i) the dependency structure proposed by XLE for a given sentence, and ii) other morphosyntactic, structural or lexical semantic features provided by the XLE grammar, as detailed in Table 1.[9]

6.1. *Dependency structure*

As detailed in Section 3, we extract labeled dependency representations from the XLE output. The labels are taken directly from the f-structure paths. We employ two

8. Available from http://www.cis.upenn.edu/∼dbikel/software.html.
The main idea in randomized parsing evaluation is that given a null hypothesis of no difference between two sets of results, shuffling the results from one system with those of the other should produce a difference in overall results equal to or greater than the original difference, since the individual scores then should be equally likely. If the performance of two sets differs significantly, on the other hand, the shuffling of the predictions will very infrequently lead to a larger performance difference. The shuffling is iterated 10,000 times and the total number of differences in results equal to or larger than the original is recorded. The relative frequency of the number of differences is then interpreted as significance of the difference.
9. A short version of these results is presented in Øvrelid *et al.* (2009).

	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
Morph	92.53	89.74	89.45	87.11
Struc	92.51	89.72	89.10	86.50
Sem	92.53	89.74	89.21	86.63
Single+Feats	92.52	89.69	90.01	87.77
Complex+Feats	92.53	89.70	**90.02**	**87.78**

Table 4. *Overall results in ten-fold cross-validation experiments using gold standard part-of-speech tags, expressed as unlabeled and labeled attachment scores (UAS/LAS)*

strategies for the extraction of dependency structures from output containing multiple heads. We attach the dependent to the closest head and i) label the dependency with the corresponding label (Single), and ii) label the dependency with the complex label corresponding to the concatenation of the labels from the multiple head attachments (Complex). In this way we preserve a part of the analysis, while outputting well-formed dependency trees.

The results for English are presented in Table 4. The addition of the proposed dependency structure from the grammar-driven parser (Single) causes a small but significant improvement of results ($p<.02$; $\sigma=.05$). In terms of labeled accuracy the results improve by 0.15 percentage points, from 89.64 to 89.79, constituting a 1.4% reduction of error rate. The introduction of complex dependency labels to account for multiple heads in the LFG output (Complex) causes a smaller improvement of results than the single labeling scheme.

The corresponding results for German are also presented in Table 4. We find that the addition of grammar-derived dependency structures with single labels (Single) improves the parse results significantly ($p<.0001$; $\sigma=0$), 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, constituting a 10.3% reduction of error rate. For the German data, we find that the addition of the dependency structure with complex labels (Complex) gives a further small, but non-significant, improvement over the experiment with single labels (Single).

6.2. *Additional grammar-derived features*

The additional linguistic features extracted from the grammar output and presented in Table 1 were added in a set of experiments for English and German. We experi-

Level	Features
Morph	CASE, DEF, DEVERBAL, FUT, GEND, GENITIVE, MOOD, NUM, PASSIVE, PERF, PERS, TENSE
Struc	AUXFLIP, AUXSELECT, CLAUSETYPE, COHERENT, COORDLEVEL, COORD, COORDFORM, GOVPREP, SUBCAT
Sem	ADJUNCTTYPE, ADVTYPE, ATYPE, COMMON, COUNT, DEIXIS, GENDSEM, HUMAN, LOCATIONTYPE, NTYPE, PROPERTYPE, PSEM, PTYPE, TIME, VTYPE

Table 5. *Features from XLE output, ordered by level of linguistic analysis – (morphosyntactic (Morph), structural (Struc) and semantic (Sem)*

mented with several feature models for the inclusion of the additional information; however, we found no significant differences when performing a forward feature selection.[10] The addition to the feature model simply adds the XFEATS of the *top* and *next* tokens of the parse configuration.

The English parse results with the addition of the grammar-extracted features in Table 1 (Feats) are presented in Table 4. We find that the results improve significantly compared to the baseline ($p<.04$; σ=.08) by 0.13 percentage points. For German, we find that the addition of all the features presented in Table 1 (Feats) causes a significant improvement over the baseline ($p<.0001$; σ=0), albeit slightly lower than the effect obtained with the addition of the dependency structure proposed by the grammar-driven parser.

As Table 1 illustrated, there are a large number of features extracted from the XLE output, and these pertain to various linguistic levels – morphosyntax, syntactic structure and semantics. A clearer understanding of the contribution of the individual features is therefore important. We performed an additional set of experiments employing the feature subsets presented in Table 5 – morphosyntactic features, such as CASE, GENDER and NUMBER, structural features, such as COORDLEVEL, SUBCAT and CLAUSETYPE, and semantic features, such as ADVTYPE, LOCATIONTYPE and TIME.

The results using feature subsets are presented in Table 4. German has a richer morphological system than English, with agreement based on morphosyntactic categories such as case and gender. For German, it is not surprising that the morphosyntactic features (Morph) give the most performance boost. It is clear, however, that the improvement observed with the full feature set does not entirely stem from morphosyntax. The inclusion of both the structural (Struc) and semantic (Sem) features

10. The feature selection experiments were performed starting from a minimal set of features for the linguistic information: S:*top*, I:*next*, and adding new features expressing more of the linguistic context (preceding and following tokens in the input and stack, information about the head, etc.) in a one-by-one fashion. We finally ran an experiment with a maximal feature model including all the tested features. No further significant improvements were observed, hence we settled for the minimal feature model.

give significant improvements over the baseline. And, finally, their combination gives the best result. For English, we find that the three feature subsets all cause slight, but not significant, improvements over the baseline and that the combination of these outperform the individual features and give significant improvements in overall results.

6.3. *Combination*

The experiments testing the addition of proposed dependency structures and additional linguistic features from the grammars showed that these individually cause significant improvements in terms of parse accuracy for both English and German. It might very well be, however, that the features contribute information which serves the same purpose and hence will not lead to an accumulative effect when combined. The results for experiments combining both sources of information – dependency structures and additional features – are presented in the final lines of Table 4.

We find that for the English parser, the combination of the features does not cause a further improvement of results, compared to the individual experiments. Rather, the results are lower than in the individual experiments. In the German experiments, on the other hand, the effect of combining the features is positive. The combined experiments (Single+Feats, Complex+Feats) differ significantly from the baseline experiment ($p<.0001$; $\sigma=0$), as well as the individual experiments – Complex ($p<.01$; $\sigma=.08$) and Feats ($p<.0002$; $\sigma=.001$) – reported in Sections 6.1-6.2. By combining the grammar-derived features we improve on the baseline by 1.81 percentage points, from labeled accuracy of 85.97 to 87.78, constituting a 12.9% reduction of error rate.

7. Error analysis

The experiments presented in the previous section show that parser combinations with large-scale LFG grammars can improve data-driven dependency parsing for both English and German, even though the level of improvement differs between the two languages. However, overall parse improvements say very little about the precise effect of the added linguistic knowledge during parsing. An in-depth error analysis was therefore performed.

We seek to compare the effects observed in the two languages; however, the annotation schemes for the two treebanks are not isomorphic. Figures 3-4 compare the F-scores for a common set of the most frequent dependency relations in the experiments adding dependency structure only (Single) and additional features only (Feats) for English and German. We examine subjects, objects, adverbials, nominal modifiers and coordinations. The treebanks differ in particular in the treatment of various adverbials. For English we included two types – ADV and TMP, whereas in the German annotation these are subsumed under the MO label.

Differences in the linguistic expression of syntactic structure in different languages clearly influence the parse performance. It is well known that certain properties of

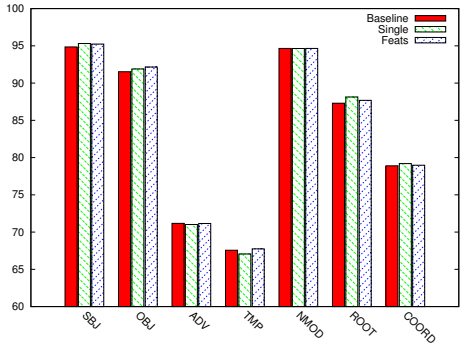

Figure 3. English: *F-scores for subjects* (SBJ)*, objects* (OBJ)*, adverbials* (ADV)*, temporal adverbials* (TMP)*, nominal modifiers* (NMOD)*, root* (ROOT) *and coordinations* (COORD) *in the Baseline, Single and Feats experiments*

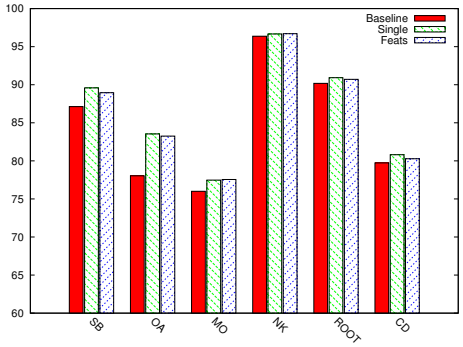

Figure 4. German: *F-scores for subjects* (SB)*, objects* (OA)*, modifiers* (MO)*, noun kernel modifiers* (NK)*, root* (ROOT) *and coordinations* (CD) *in the Baseline, Single and Feats experiments*

German, such as variable argument placement and case syncretism, pose additional challenges in parsing (Kübler *et al.*, 2006).[11] We may note first of all, that the performance for argument relations, such as subjects, objects and predicatives, is notably higher for English than for German. For instance, the baseline parser for English obtains an F-score of 94.85 for the subject relation (SBJ), whereas the corresponding German parser gets 87.12 (SB). In this respect there is clearly also more room for improvement in the German analyses.

We may furthermore note that the English parser has a lower performance for adverbial relations (ADV,TMP) than the German parser (MO). The dependency relations employed in the English treebank distinguish a more fine-grained set of adverbial relations, including temporal (TMP), directional (DIR) and locative (LOC) adverbials, where the German treebank groups these together under one dependency relation (MO). This is clearly an example of the granularity of the tag set making parsing of adverbials in English a harder task. The addition of grammar-derived structures does not improve on the analysis of adverbials to any large extent.

Generally, we can say that the effects observed with the addition of structure and features from the XLE grammars to the treebanks depend on the "division of labor" between the two. In particular, the effect depends on where the grammars may con-

11. Case syncretism denotes the situation where one inflected form corresponds to several cases. For instance, German does not have separate forms for nominatives and accusatives in the feminine gender.

		Freq	Base	Best
SBJ	subject	0.09	94.85	95.31
ROOT	root	0.05	87.30	88.13
P	punc	0.01	69.23	72.59
OBJ	object	0.06	91.53	91.90
PRD	predicative	0.02	86.50	87.28
COORD	coordination	0.03	78.90	79.19
AMOD	adjectival mod.	0.02	76.93	77.41
SUB	subord	0.01	90.33	90.95
PMOD	prep. mod.	0.11	95.97	96.03
APPO	apposition	0.02	75.26	75.53

Table 6. *Top 10 improved dependency relations in the English Single experiment, ranked by their weighted difference of balanced F-scores*

		Freq	Base	Best
MO	modifier	0.14	76.00	77.99
OA	acc. obj.	0.04	78.04	84.31
SB	subject	0.08	87.12	90.01
AG	genitive attr.	0.03	82.93	90.78
NK	nom. mod.	0.35	96.37	96.85
DA	dative	0.01	52.85	73.26
OC	clausal obj.	0.05	88.85	90.39
CJ	conjunct	0.04	72.32	74.20
ROOT	root	0.06	90.18	91.22
MNR	postnom. mod.	0.03	67.86	69.63

Table 7. *Top 10 improved dependency relations in the German Complex+Feats, ranked by their weighted difference of balanced F-scores*

tribute with generalizations which are not made explicitly in the treebank data set. Tables 6-7 show ranked lists of the dependency relations for which the parser performance improves the most in the best performing systems for English and German, respectively.[12] If we compare the effects in the two languages, we may note several points of difference.

Whereas we observe a general improvement for argument relations, such as subjects and objects, in both systems, we find that the analysis of adverbials improves to a larger extent for German. The modifier relation MO which is employed largely for prepositional phrases at the sentence level is one of the relations for which parser performance improves the most in the German experiments. This may in part be traced back to the difference in the annotation schemes and grammars for the two languages. The MO relation in the Tiger treebank is employed for all types of modifiers at the sentence level, as well as nominal adverbs at the phrase level. The German XLE grammar, however, makes some finer distinctions and distinguishes general adjuncts (ADJUNCT) from various oblique arguments (OBL-DIR, OBL-LOC). For instance, the German example in (5) is a sentence where the baseline parser erroneously attaches the prepositional phrase *auf den Glastüren* 'from the glass doors' to the preposition *als* 'as/like' and assigns it the MNR-relation. The XLE analysis, however, attaches it correctly to the verb *erscheinen* 'seem' and analyzes it as an oblique argument

expressing direction (OBL-DIR), whereby the stacked parser subsequently performs correct attachment and labeling.

(5) *. . . nicht als Spiegelbilder* **auf** *den Glastüren des Einkaufspalastes erscheinen*
. . . not as reflections from the glass-doors the shopping-centres.GEN seem
'. . . do not seem like reflections coming from the glass doors of the shopping malls'

As mentioned earlier, the English treebank annotation makes a range of adverbial distinctions, whereby these are largely annotated as ADJUNCT by the XLE-grammar. The use of the total set of additional XLE features for English benefits argument relations, (subjects, objects, predicatives), but also the temporal adverbial relation (TMP). The features TIME for nouns, as well as the distinctions made by the ADVTYPE feature (sentence adverbial or vp-adverbial) contribute to this improvement.

Where the annotations in the grammar and treebank do not differ, the added predictions serve to supply extra evidence in the parsing of "hard cases". For instance, we find that subjects are one of the top most improved dependency relations for the English parser: see Table 6. The improved instances are largely analyzed as subjects by the XLE grammar as well, such as the subject of the subordinate clause with missing complementizer in (6):

(6) Mr. Sulzberger said the scheduled **opening** of . . .

A final point in the comparison of the ranked lists in Tables 6-7 is that the analysis of punctuation is improved in the English experiments, but not in the German. This is once again explained by differences in the grammars, where the English grammar is better at handling sentence-final punctuation.[13]

We noted earlier that the dependency analyses proposed by the grammar and the treebanks are not always identical, and we have not done any modifications in order to make them more similar. A question is whether systematic differences in the analyses actually contribute to the observed improvements. For instance, we have seen that the LFG analysis systematically analyzes the lexical verb as the root of the sentence,

12. In order to summarize improvement with respect to dependency relation assignment when comparing two parsers, we rank the relations by their frequency-weighted difference of F-scores. For each dependency relation, the difference in F-scores is weighted by the relative frequency of the dependency relation, $\frac{Deprel}{\sum_i Deprel_i}$, in the treebank.

13. The CoNLL evaluation script does not take into account punctuation during overall scoring, hence the overall results expressed as unlabeled and labeled attachment scores are calculated disregarding punctuation. The improvement in overall results for English when scoring for punctuation is slightly higher: 0.18 percentage points, as compared to 0.15 without punctuation. In calculating performance per dependency relation, however, we may look specifically at punctuation.

whereas the treebanks make the finite verb the root. If we look at the relations for which we observe improvements in the two languages, we find that these exhibit a relatively large degree of mismatch in terms of head assignment. For instance, the grammar-driven analysis of subjects does not match the treebank analysis in terms of head assignment for 54% and 45% of the instances in the English and German data sets, respectively. The same holds for many other improved relations, such as OBJ (57% mismatch) and ROOT (27% mismatch) for English, and MO (47% mismatch) and OA (28% mismatch) for German. Another point of mismatch is found in labeling, where one treebank label may often correspond to several different labels in the grammar analysis. Whereas there is usually one label in the majority, we find that the German MO-relation discussed above, for instance, corresponds to the grammar-based ADJCT-relation in 72% of the training instances and to other labels such as OBL-LOC, OBL-DIR, OBJ in the rest. This indicates that the improvement observed does not rely on identical analyses from the resources employed.

7.1. *Parser quality*

The output from the grammar-driven parser is necessarily noisy and the coverage is not one hundred percent. Even so, the parsing experiments show that the data-driven parser may generalize over the input and actually acquire improved linguistic analyses. As mentioned in Section 2, we run the parser in so-called fragmented mode, where sentences which do not receive a full analysis by the parser are simply bundled together under a special root node denoted 'FRAGMENTS'. Such fragmented nodes typically consist of lower-level substructures such as noun phrases and prepositional phrases which are usually connected in a sentential analysis. One might ask how important the quality of the output from the grammar-driven parser is. In our original setup we chose to prioritize coverage. However, it might be that the additional fragmented information is simply of such poor quality as to cause more damage than good during parsing.

In a series of experiments we investigate the influence of the parse quality further by training a baseline and a stacked parser exclusively on sentences that receive a full analysis from the grammar-driven parser. For English, this results in a loss of 15.8% of the sentences. The German parser apparently outputs more fragmented parses, and 30% of the total sentences are excluded in these experiments. For the stacked parsers, we employ the settings for the best systems in the earlier experiments, i.e. the Single system for English and the Complex+Feats system for German, the only difference being the amount and quality of the training data, which is restricted to non-fragmented sentences.

The first two rows of Table 8 present the results obtained in the experiments with non-fragmented parse features. Compared to the new baseline consisting of non-fragmented sentences parsed with the baseline parser (Baseline$_{NonFrag}$), we observe a comparatively somewhat larger improvement for both languages: 0.24 percentage points for English or 2.4% reduction of error rate, compared to the earlier 0.15 or

	English		German	
	UAS	LAS	UAS	LAS
Baseline$_{NonFrag}$	92.48	89.63	89.55	86.89
Best$_{NonFrag}$	92.71	89.87	91.09	89.04
Baseline	92.48	89.64	88.68	85.97
Best$_{FragSpec}$	92.61	89.79	90.02	87.78
Best$_{FragUnSpec}$	92.55	89.70	89.81	87.54

Table 8. *Overall results in ten-fold cross-validation experiments with non-fragmented sentences only (NonFrag) and with specified (FragSpec) vs. unspecified fragmented input (FragUnSpec), expressed as unlabeled and labeled attachment scores (UAS/LAS)*

1.4% error rate reduction, and 2.15 percentage points or 15.3% reduction in error rate for German, compared to the earlier 1.81 or 12.9% error rate reduction. This is not surprising, as these experiments provide us with a test of the performance possible under near-perfect conditions, without having to resort to the gold standard of the LFG analyses. In an error analysis, we note that the main trends of improvement in terms of dependency relations are the same for both languages, only somewhat stronger.

The experiments with non-fragmented data do not represent a realistic setting, as we are training on a subset of the training data, hence possibly overlooking phenomena that are problematic for both parsers. An alternative strategy would be to include additional features only for non-fragmented analyses, while still training on the whole data set. This would involve leaving a subset of the training data unspecified for its grammar-driven analysis. We ran this experiment and the results are presented in the last row of Table 8, where we have included the baseline (Baseline), as well as the best results obtained in the earlier experiments using fragmented input (Best$_{FragSpec}$), as discussed in Section 6 above. We find that the parsers trained with unspecified fragmented analyses perform worse than the fully specified parsers. The results show that coverage is important in a realistic setting and that the data-driven parser may generalize successfully over fragmented analyses.

8. Automatically assigned PoS-tags

The recent CoNLL shared tasks (Buchholz and Marsi, 2006; Nivre *et al.*, 2007) have provided benchmarks within the area of data-driven dependency parsing. The general experimental methodology employed in these tasks has involved training and testing on gold standard parts of speech. Although this allows one to focus solely on the task of syntactic parsing without interference from tagging errors, it is also a somewhat unrealistic setting for further application and the results may give an over-optimistic view of the accuracy that can be expected when parsing new text. It is there-

	English		German	
	UAS	LAS	UAS	LAS
Baseline$_{AUTO}$	91.67	88.56	87.13	84.14
XPos	91.46	88.26	86.72	83.61
DepStruc (Single/Complex)	**91.88**	**88.77**	87.36	84.69
Feats	91.84	88.77	87.15	84.41
DepStruc+Feats	**91.88**	**88.76**	87.86	85.31
DepStruc+Feats+XPos	91.63	88.44	**88.27**	**85.72**

Table 9. *Overall results in ten-fold cross-validation experiments using automatically assigned part-of-speech tags, expressed as unlabeled and labeled attachment scores (UAS/LAS)*

fore important to show how the effects resulting from the use of the grammar-derived dependency structures and other features are affected by the use of automatically assigned part-of-speech tags.

In order to test the parser stacking approach with automatically assigned part-of-speech tags, we tag the treebanks using the TreeTagger (Schmid, 1994), which has models for both English and German.[14] We simply replace the gold standard part-of-speech tags in the treebanks with the automatically assigned tags, and subsequently retrain and retest the parsers. The experimental setup is otherwise identical to the previous experiments described in Section 6 above. As is to be expected, the baseline results are lower with the automatically assigned tags: see the results for the Baseline$_{AUTO}$ parsers in Table 9. We may furthermore note that the deterioration in parse results compared to the baseline employing gold standard tags is larger for German (1.83 percentage points LAS) than English (1.08 percentage points LAS). As mentioned earlier, XLE comes with its own tokenizer and part-of-speech tagger, hence may be applied directly to raw text. It employs a somewhat different part-of-speech tag set than the one employed by TreeTagger, so one interesting question is clearly whether the XLE output may benefit the parser also in this case. An experiment where the XLE PoS-tags were added as features for the parser is reported in the second line of Table 9 (XPos).[15] We find that the use of a second set of part-of-speech tags actually causes a deterioration in results for both languages. It is clear that the PoS-tag sets do not complement each other in a way that is useful in the parser combination setting.

We also perform experiments equivalent to the ones reported in Section 6, where we were using gold standard PoS-tags in the parse models. As before, we find that the

14. TreeTagger is available from http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger

15. The XLE part-of-speech tags were included for the same tokens of the parse configuration as the TreeTagger, and previously, the gold standard, part-of-speech tags: see Section 5.2 above.

use of both the grammar-derived dependency structures (DepStruc[16]) and the other linguistic features (Feats) during parsing causes significant improvements for both languages. For English we find that the added information has a slightly more pronounced effect than in the gold standard experiments, and both the addition of dependency structure (DepStruc) and additional features (Feats) causes significant improvements ($p<.01;\sigma=.06/03$). The German results show that the addition of both dependency structure ($p<.002;\sigma=.01$) and additional features ($p<.05;\sigma=.01$) improves parse results and we also observe additional improvements from their combined effect ($p<.0001;\sigma=0$). The effects are somewhat less pronounced than in the gold standard experiments, as is to be expected.

The effects observed in the experiments using automatically assigned part-of-speech tags largely corroborate those noted in Section 6 above. For English, we find that it is the addition of dependency structure that has the largest effect on results (0.21 percentage points labeled improvement or 1.8% reduction of error rate). It is also clear, however, that the linguistic features (Feats) are contributing more in a setting where the part-of-speech tags are less reliable.[17] When performing an error analysis for these results, we find that the results are very similar to those reported in the previous section, and also in terms of specific improvement at the level of dependency relations. For English, we observe improvements for argument relations such as subjects (SBJ) and objects (OBJ), as well as functional categories such as dependency root (ROOT) and punctuation (P). As before, the analysis of various adverbial modifiers does not improve compared to the baseline.

In the case of the German experiments, we also find a very similar pattern of results. The combination experiments achieve the highest results, where it seems very much to be the case that the more features, the better. The combination of dependency structures with additional features (DepStruc+Feats) provides an improvement of 1.17 percentage points in overall results. The further addition of the XLE part-of-speech tags, which did not provide any benefit on their own (XPos), cause a further improvement of 1.58 percentage points or 10% reduction of error rate, compared to the baseline. A more detailed error analysis shows that the effects obtained with automatically assigned tags are somewhat shifted. The added information has the largest effect on the assignment of argument relations – objects (OA), genitive attributes (AG) and subjects (SB) – whereas the aforementioned effect on the adverbial relation (MO) is somewhat less pronounced.[18]

16. Building on the experiments using gold standard tags reported in Section 6, we use single labels for English and complex labels expressing multiple head assignments for German.

17. Unlike the results obtained using gold standard tags, however, we find that the combination of dependency structure and additional features does not cause a deterioration of results, compared to the addition of grammar-derived dependency structures only.

18. The performance gain observed for the MO-relation in the German experiment with automatic PoS-tags is 1.19 percentage points improvement in F-score, compared to 1.99 in the gold standard experiment: see Table 7.

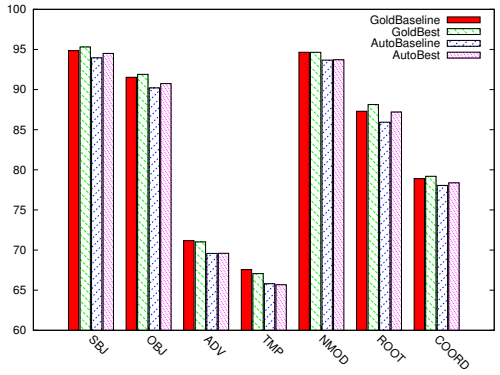

Figure 5. English: *F-scores for subjects (*SB*), objects (*OA*), modifiers (*MO*), noun kernel modifiers (*NK*), root (*ROOT*) and coordinations (*CD*) in the gold standard baseline (GoldBaseline) and best (GoldBest) experiments, compared with automatic tag assignment baseline (AutoBaseline) and best (AutoBest)*

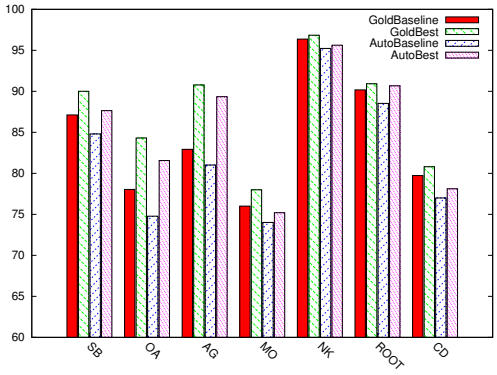

Figure 6. German: *F-scores for subjects (*SB*), objects (*OA*), genitive attr. (*AG*), modifiers (*MO*), noun kernel modifiers (*NK*), root (*ROOT*) and coordinations (*CD*) in the gold standard baseline (GoldBaseline) and best (GoldBest) experiments, compared with automatic tag assignment baseline (AutoBaseline) and best (AutoBest)*

As in the earlier experiments, we found that the effects of our features were more pronounced for German. In fact, the error analysis shows that for some dependency relations the performance with automatic tags actually reaches baseline gold standard performance or even better. However, it goes for both languages that the use of grammar-derived features during data-driven parsing, at least partially, circumvents the deterioration of results when moving to applications to raw text. Figures 5-6 show comparisons of the performance (F-score) obtained for the most frequent set of dependency relations in the gold standard baseline and best experiments, as well as the automatic baseline and best experiments.[19] In general we find that the parser stacking setup is an effective means for augmenting the simple part-of-speech-tagging/parsing pipeline with the syntactically informed category decisions of the full grammar-driven parser.

19. The best results in the gold standard and automatic experiments for English were the Single and DepStruc experiments, respectively, and for German, the Complex+Feats and DepStruc+Feats+XPos, respectively.

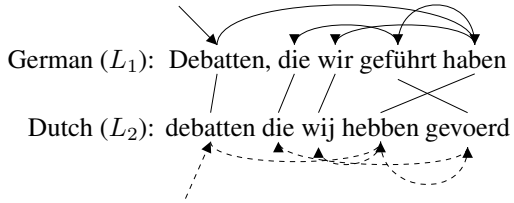

Figure 7. *Dependency tree projection from German to Dutch*

9. Parser stacking in crosslingual projection

In order to assess the usefulness of the combined parsers, we also provide an application-oriented evaluation. In particular, we compare the baseline parsers and the combined parsers in terms of the performance that is achieved when the respective parsers are used as source parsers in a framework for the crosslingual projection of syntactic annotations (Hwa *et al.*, 2005; Spreyer and Kuhn, 2009).

Crosslingual annotation projection (Yarowsky *et al.*, 2001) induces linguistic annotations for a target language by exploiting word-aligned parallel corpora in combination with existing resources for another language (the source language). It thus avoids the expensive annotation process of manual treebank creation for the target language. To be more concrete, in the context of dependency parsing we take advantage of the fact that state-of-the-art parsers exist for a handful of languages, and parallel text is available (e.g., Koehn (2005)) for which word alignments can be established automatically (Och and Ney, 2003). Thus, given a parallel corpus with dependency annotations in the source language, the dependencies can be projected along the word alignment to the target language. This is illustrated in Figure 7, with German as the source language and Dutch as the target language. For example, the edge (*haben*, *geführt*) in the German source parse is projected to (*hebben*, *gevoerd*) in Dutch, via the word alignments *haben*↔*hebben* and *geführt*↔*gevoerd*. Note that the notion of parallelism that is exploited here is independent of word order, given the word alignment. This can be seen in the verbal complex in Figure 7, where the main verb precedes the auxiliary in the German relative clause, while the opposite is true for Dutch. However, this difference is contained in the word alignment, and the projected dependency relations form an appropriate (non-projective) dependency tree for the Dutch sentence. Note that in many cases, the result of projecting a dependency tree based on an automatic word alignment is not a fully connected dependency graph. However, we also explore the usability of the resulting dependency parse fragments.[20]

20. It is important to note that the dependency parse fragments in the present section are distinct from the fragmented XLE parses discussed above.

Using the baseline parser and the best combined German system from Section 6 as alternative source parsers, we parsed the German portions of 100,000 parallel sentences from the Europarl corpus (Koehn, 2005). We subsequently projected the trees to the Dutch translations as explained above. Discarding those Dutch parses that do not form trees (i.e., fragmented parses), this yielded data sets of 15,300 (baseline) and 15,600 (combined) words when projecting from German. We then used the projected trees as training data for MaltParser.

When fragmented projected parses are discarded, the distribution of the remaining data is highly skewed and most non-trivial examples are lost. This is because the direct correspondence assumption (Hwa *et al.*, 2005) does not hold in general: Although languages do tend to exhibit astonishing degrees of parallelism, translations are rarely completely isomorphic. Hence, only sentence pairs with a one-to-one correspondence between the words in the source and target language will receive a connected parse through projection. We thus employ a slightly modified version of MaltParser which is capable of successfully learning from fragmented input (Spreyer and Kuhn, 2009), and can therefore make use of the full set of projected dependencies.

Table 10 shows the UAS of the projected parsers, evaluated on the CoNLL06 test set (Buchholz and Marsi, 2006) of 386 sentences from the Alpino treebank (van der Beek *et al.*, 2002). We see that the combined parser significantly[21] outperforms the baseline in both training regimens (on trees as well as on fragments).

We use the crosslingual projection set-up in order to assess whether the increase in parsing performance observed in the standard, in-domain gold standard treebank evaluation will have a favorable effect on out-of-domain parser application under realistic application conditions: the German stacked parsers are used outside their training domain, namely on Europarl data. The Europarl-trained (projected) Dutch parser is then evaluated on the Dutch Alpino treebank gold standard. Note the relatively high number of potential noise sources in this set-up: the German (baseline) data-driven parser is applied on raw out-of-domain text; the grammar-driven parser (in parser stacking) is run fully automatically, with its own robustness techniques (skimming and fragment parsing); the word alignment is automatic; data-driven training of the Dutch target parser is on fragmented crosslingual data from the Europarl domain, but the final evaluation is on Dutch gold standard data from a different domain. The fact that the parser stacking improvements observed under laboratory conditions carry over to this set-up indicates that it is a robust effect.

21. A cross-validation scheme is not applicable with a monolingual test set. Further complication arises from the fact that the underlying source language parsers not only differ in terms of accuracy, but indirectly lead to projected training sets that do not necessarily contain the same sentences. This is because different parse trees may be fragmented differently when projected to the target language, and fragmentation is the criterion for the training data selection. We therefore perform significance testing using the t-test ($p \ll 0.01$) over the results of training on ten random permutations of the respective training data. In Table 10 we report the means and standard deviations of these results.

	Trees	Fragments
Projection from baseline	63.94 (0.70)	67.67 (0.39)
Projection from stacked	**65.38 (0.29)**	**68.60 (0.40)**

Table 10. *Mean unlabeled accuracy (UAS) and standard deviations for Dutch parsers projected from German*

10. Discussion and conclusion

The idea of combining several parsers is certainly not new. Work on parser ensembles usually makes use of a voting strategy of some kind in order to derive a single prediction for an output (Sagae and Tsuji, 2007; Zhang *et al.*, 2008). As Nivre and McDonald (2008) point out, the parser stacking differs from these in the integration of the parsers during *learning* of the parse models. The work described here uses the technique employed in Nivre and McDonald (2008) for integrating graph-based and transition-based dependency parsers and extends it by employing features taken from a grammar-driven parser. Like Nivre and McDonald (2008), we supply a data-driven dependency parser with features from a different parser. 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 outputting a deep LFG analysis. We also show how a range of other features – morphological, structural and semantic – from the grammar-driven analysis may easily be employed during data-driven parsing and lead to significant improvements in parse results.[22] The previous work which most resembles this is the use of grammar-derived features from HPSG grammars in data-driven dependency parsing (Zhang and Wang, 2009). They show that parser combination is beneficial for domain-adaptation in English and argue that the grammar-driven parser provides domain-independence.

The approach detailed here should be easily applicable to grammars written within different theoretical frameworks. This opens a range of interesting possibilities, both in terms of combining parsers from other theoretical frameworks and of generalization of the method to other languages. It is clearly common for many languages to have grammars hand-written in different theoretical frameworks and with varying coverage and quality. The approach to parser stacking presented in this paper creates the possibility of combining these in a way that makes use of the generalizations expressed by the hand-written grammar within a framework which has furthered the state of the art for a range of languages. The requirements for application to a new language are

22. English was not among the languages investigated in Nivre and McDonald (2008). Our best results for German, combining dependency structures and additional features, are slightly higher than those reported for MaltParser (0.11 percentage points). These results are not, however, directly comparable as they were obtained on different test sets.

the existence of a syntactic treebank on which to train a parser, one or more grammars for the language and a mapping from the grammar annotation to dependency representations. There are clear extensions to the work presented here that constitute future plans for work. Some main themes in this research would look at improved data-driven modeling through richer features to take into account the full potential of the deep resources/grammars, the effect of different theoretical frameworks and their representations of phenomena such as long-distance dependencies, raising/control, etc., and the effect of other types of information, such as semantic and discourse-related analyses provided by the deep grammars. The work presented here provides an important contribution to this.

This paper has presented systematic 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 also presented a set of additional, linguistic 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 order to address the question of whether the effect is merely due to the combination of two independently developed parses or whether we indeed see a substantial effect of the two complementary parsing approaches, we have performed detailed analyses and further experiments.

A feature analysis through feature subset experiments indicates that information from several linguistic levels – morphosyntactic, structural and semantic – contributes to the observed effect and does so in a way which reflects properties of the languages under analysis. An in-depth error analysis has shown how the effects of the added features rely on the combined distinctions expressed in the treebank and the grammars, as well as systematic mismatches between the two.

Experiments assessing the importance of parser quality have indicated that a slight performance gain would be possible with a better parser. More importantly, however, the experiments have also shown that coverage is important and that the data-driven parser is capable of generalizing over quite noisy input. In application to raw text, the effect of automatically assigned part-of-speech tags is an important factor. Here, we have shown that the use of grammar-derived information in data-driven dependency parsing helps reduce the deterioration of results which accompanies application-realistic settings and thus provides a means for augmenting the standard part-of-speech-tagging/parsing pipeline with the syntactically informed category decisions of the full grammar-driven parser. We have furthermore provided results from application to the task of crosslingual annotation projection. We found that in a setting with a great deal of noise, where the parsers are applied to out-of-domain data, the effect of parser stacking remains and provides significant improvement of results.

Acknowledgments

The authors would like to thank Joakim Nivre for useful discussions about parser modifications and feature models for parser stacking, as well as the anonymous reviewers for insightful and helpful suggestions and comments. The work reported in this paper was supported by the Deutsche Forschungsgemeinschaft (DFG; German Research Foundation) in (i) the Emmy Noether project PTOLEMAIOS, on Grammar Induction from Parallel Corpora, and (ii) SFB 632 on Information Structure, project D4 (Methods for interactive linguistic corpus analysis).

11. References

Bod R., *Beyond Grammar: An experience-based theory of language*, CSLI Publications, Stanford, CA, 1998.

Brants S., Dipper S., Eisenberg P., Hansen-Schirra S., König E., Lezius W., Rohrer C., Smith G., Uszkoreit H., "TIGER: Linguistic Interpretation of a German Corpus", *Research on Language and Computation*, vol. 2, p. 597-620, 2004.

Buchholz S., Marsi E., "CoNLL-X Shared Task on Multilingual Dependency Parsing", *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, p. 149-164, 2006.

Butt M., Dyvik H., King T. H., Masuichi H., Rohrer C., "The Parallel Grammar Project", *Proceedings of the COLING-2002 Workshop on Grammar Engineering and Evaluation*, p. 1-7, 2002.

Cahill A., Burke M., O'Donovan R., Riezler S., van Genabith J., Way A., "Wide-Coverage Deep Statistical Parsing using Automatic Dependency Structure Annotation", *Computational Linguistics*, vol. 34, n° 1, p. 81-124, 2008a.

Cahill A., Maxwell J. T., Meurer P., Rohrer C., Rosen V., "Speeding up LFG parsing using C-structure pruning", *Proceedings of the Workshop on Grammar Engineering Across Frameworks*, p. 33-40, 2008b.

Chang C.-C., Lin C.-J., "LIBSVM: A Library for Support Vector Machines", 2001, Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Clark S., Curran J. R., "Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models", *Computational Linguistics*, vol. 33, n° 4, p. 493-552, 2007.

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

Forst M., Crysmann B., Fouvry F., Hansen-Schirra S., Kordoni V., "Towards a dependency-based gold standard for German parsers – The TiGer Dependency Bank", *Proceedings of the COLING Workshop on Linguistically Interpreted Corpora (LINC'04)*, p. 31-38, 2004.

Hockenmaier J., Steedman M., "Generative Models for Statistical Parsing with Combinatory Grammars", *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 335-342, 2002.

Hwa R., Resnik P., Weinberg A., Cabezas C., Kolak O., "Bootstrapping Parsers via Syntactic Projection across Parallel Texts", *Natural Language Engineering*, vol. 11, n° 3, p. 311-325, 2005.

Johansson R., Nugues P., "Extended constituent-to-dependency conversion for English", *in* J. Nivre, H.-J. Kaalep, M. Koit (eds), *Proceedings of NODALIDA 2007*, p. 105-112, 2007.

Kaplan R., Riezler S., King T. H., Maxwell J. T., Vasserman A., Crouch R., "Speed and accuracy in shallow and deep stochastic parsing", *Proceedings of HLT-NAACL*, p. 97-104, 2004.

Klein D., Manning C. D., "Accurate Unlexicalized Parsing", *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 423-430, 2003.

Koehn P., "Europarl: A Parallel Corpus for Statistical Machine Translation", *Proceedings of the MT Summit 2005*, 2005.

Kübler S., Hinrichs E., Maier W., "Is it really that difficult to parse German?", *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 111-119, 2006.

Marcus M. P., Santorini B., Marcinkiewicz M. A., "Building a large annotated corpus for English: The Penn Treebank", *Computational Linguistics*, vol. 19, n° 2, p. 313-330, 1993.

Miyao Y., Tsujii J., "Probabilistic disambiguation models for wide-coverage HPSG parsing", *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 83-90, 2005.

Nilsson J., Nivre J., "Pseudo-projective dependency parsing", *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 99-106, 2005.

Nivre J., "An Efficient Algorithm for Projective Dependency Parsing", *Proceedings of the Eighth International Workshop on Parsing Technologies*, p. 149-160, 2003.

Nivre J., Hall J., Kübler S., McDonald R., Nilsson J., Riedel S., Yuret D., "CoNLL 2007 Shared Task on Dependency Parsing", *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, p. 915-932, 2007.

Nivre J., Hall J., Nilsson J., "MaltParser: A Data-Driven Parser-Generator for Dependency Parsing", *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, p. 2216-2219, 2006a.

Nivre J., McDonald R., "Integrating Graph-Based and Transition-Based Dependency Parsers", *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, p. 950-958, 2008.

Nivre J., Nilsson J., Hall J., Eryiğit G., Marinov S., "Labeled pseudo-projective dependency parsing with Support Vector Machines", *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, p. 221-225, 2006b.

Och F. J., Ney H., "A Systematic Comparison of Various Statistical Alignment Models", *Computational Linguistics*, vol. 29, n° 1, p. 19-51, 2003.

Øvrelid L., Kuhn J., Spreyer K., "Improving data-driven dependency parsing using large-scale LFG grammars", *Proceedings of the Annual Meeting for the Association for Computational Linguistics (ACL) (Short Paper)*, p. 37-40, 2009.

Øvrelid L., Nivre J., "When word order and part-of-speech tags are not enough – Swedish dependency parsing with rich linguistic features", *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, p. 447-451, 2007.

Postal P., Ross J. R., "¡Tough Movement Sí, Tough Deletion, No!", *Linguistic Inquiry*, vol. 2, p. 544-546, 1971.

Riezler S., King T., Kaplan R., Crouch R., Maxwell J. T., Johnson M., "Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques",

Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL), p. 271-278, 2002.

Rosenbaum P., *The Grammar of English Predicate Complement Constructions*, MIT Press, Cambridge, 1967.

Sagae K., Tsuji J., "Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles", *Proceedings of Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*, p. 1044-1050, 2007.

Schmid H., "Probabilistic part-of-speech tagging using decision trees", *Proceedings of the International Conference on New Methods in Language Processing*, p. 44-49, 1994.

Spreyer K., Kuhn J., "Data-Driven Dependency Parsing of New Languages Using Incomplete and Noisy Training Data", *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, Association for Computational Linguistics, Boulder, CO, p. 12-20, June, 2009.

van der Beek L., Bouma G., Malouf R., van Noord G., "The Alpino dependency treebank", *Computational Linguistics in the Netherlands (CLIN)*, p. 8-22, 2002.

Yarowsky D., Ngai G., Wicentowski R., "Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora", *Proceedings of HLT 2001*, p. 1-8, 2001.

Zhang Y., Oepen S., Carroll J., "Efficiency in Unification-Based N-Best Parsing", *Proceedings of the 10th International Conference on Parsing Technologies*, p. 48-59, 2007.

Zhang Y., Wang R., "Cross-Domain Dependency Parsing Using a Deep Linguistic Grammar", *Proceedings of ACL-IJCNLP 2009*, p. 378-386, 2009.

Zhang Y., Wang R., Uszkoreit H., "Hybrid Learning of Dependency Structures from Heterogeneous Linguistic Resources", *Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL 2008)*, p. 198-202, 2008.