Training Parsers on Partial Trees: A Cross-language Comparison

Kathrin Spreyer*, Lilja Øvrelid*, Jonas Kuhn*†

*Department of Linguistics
University of Potsdam
Germany
{spreyer,ovrelid}@uni-potsdam.de

†Institute for Natural Language Processing (IMS)
University of Stuttgart
Germany
jonas.kuhn@ims.uni-stuttgart.de

Abstract

We present a study that compares data-driven dependency parsers obtained by means of annotation projection between language pairs of varying structural similarity. We show how the partial dependency trees projected from English to Dutch, Italian and German can be exploited to train parsers for the target languages. We evaluate the parsers against manual gold standard annotations and find that the projected parsers substantially outperform our heuristic baseline by 9–25% UAS, which corresponds to a 21–43% reduction in error rate. A comparative error analysis focuses on how the projected target language parsers handle subjects, which is especially interesting for Italian as an instance of a pro-drop language. For Dutch, we further present experiments with German as an alternative source language. In both source languages, we contrast standard baseline parsers with parsers that are enhanced with the predictions from large-scale LFG grammars through a technique of parser stacking, and show that improvements of the source language parser can directly lead to similar improvements of the projected target language parser.

1 Introduction

Annotation projection on parallel corpora has received considerable attention in NLP, as it can reduce the resource bottleneck for lesser studied languages. For syntactic parsing, dependency structures can in principle be projected in a straightforward way given a word alignment, even if there are structural differences between the source language (SL) and target language (TL). For instance, word order differences (within verbal phrases or in the relative order of adjectives and nouns) cause no harm under perfect word alignment. More critical are cases where single tokens in one language correspond to syntactic combinations in the other language.

In order to assess the practical usefulness of this approach under realistic circumstances, i.e., with automatic parser output as the projection source and a statistical word alignment, systematic comparisons are needed. We present a study that compares a particular projection approach, combined with a strategy for exploiting partially labeled sentences, for three languages of varying structural similarity with the SL: We project English dependency parses to Dutch, Italian, and German (Section 2).

Projected trees tend to be incomplete in the sense that edges may be missing, due to missing word alignments or non-parallelism of the translations. We use fMalt (Spreyer and Kuhn, 2009) to train TL parsers despite this fragmentation. fMalt is a variation of MaltParser (Nivre et al., 2006) modified to handle incomplete training data (Section 3).

We evaluate the parsers against gold standard treebank data and find that the projected parsers reduce the baseline error by 21–43% (Section 4). The error analysis focuses on how the projected TL parsers handle subjects by comparing their distribution at various stages of the projection and parsing procedure (Section 5).

For Dutch, we further present experiments with German as an alternative source language. In both SLs, we contrast standard baseline parsers with parsers that are enhanced with the predictions from large-scale LFG grammars through a technique of parser stacking, inspired by the work on combining dependency parsers in Nivre and McDonald (2008). Our experiments show that improvements thus obtained in the source parser can directly lead to similar improvements of the projected TL parser (Section 6). We compare our approach to related work in Section 7 and conclude in Section 8.

2 Projection of Dependency Trees

Most state-of-the-art parsers for natural languages are data-driven and depend on labeled training data, but manual treebank creation is expensive. It can be avoided by labeling the data automatically using *annotation projection* (Yarowsky et al., 2001): Given a dependency parser for SL, and a word-aligned parallel corpus of SL and TL, we parse the SL portion of the corpus and copy (or *project*) the dependencies to the corresponding (i.e., aligned) TL elements. This is illustrated in Figure 1 with English as the SL and Dutch, German and Italian as TLs. The links between SL and TL indicate the word alignment. We project the English trees to the TL by postulating edges between TL words (e.g., *de* and *notulen* in Figure 1a) if there is an edge between their respective English counterparts (*the* and *minutes*).

Annotation projection assumes *direct correspondence* (Hwa et al., 2005), which holds in many cases (e.g., Figure 1a), but not in general: non-parallelism between SL and TL

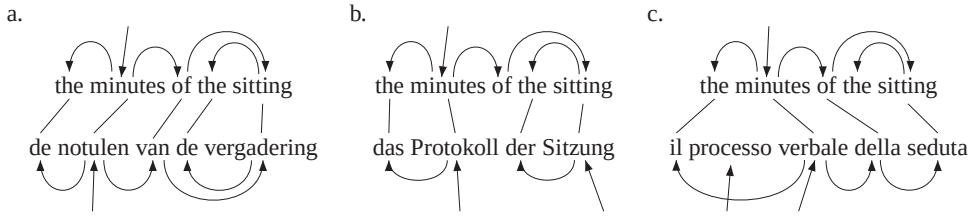

Figure 1: Dependency tree projection from English to a. Dutch, b. German, and c. Italian.

lang.	words/sent	words/frag	frags/sent
nl	24.92	1.81	13.74
de	25.27	1.83	13.78
it	27.11	2.07	13.13

Table 1: Fragmentation in the projected dependencies.

expressions causes errors or gaps in the target annotations. In Figure 1b, for instance, the possessive relation between *minutes* and *sitting* is expressed analytically with a preposition in English, but by way of genitive case marking on the NP *der Sitzung* in German. As a result, the projected structure is fragmented. The automatic alignment constitutes a further error source, as witnessed in Figure 1c: The nominal compound *processo verbale* (lit.: *spoken proceedings*) is misaligned under the intersective alignment, with the modifier *verbale* rather than the head *processo* aligned to the English *minutes*. This results not only in a fragmented, but an incorrect parse.

Note that we neither attempt to amend imperfect correspondences, nor do we discard fragmented parses. This is in contrast to related approaches to annotation projection which resort to heuristics to enforce coherence in the projected structures (cf. Section 7). We show in Section 3 that even partial annotations constitute valuable training data. Moreover, the absence of language-specific patches facilitates the analysis of the plain projections from a cross-linguistic perspective.

We use Europarl as the parallel corpus, word-aligned with GIZA++ (Och and Ney, 2003).[1] The TL texts were lemmatized and POS-tagged with TreeTagger (Schmid, 1994) The English text was processed with TreeTagger and then parsed with the baseline MaltParser of Øvrelid et al. (2009), which is trained on the dependency-converted WSJ part of the Penn Treebank (Marcus et al., 1993). The parser achieves 92% UAS (89% LAS) when gold POS-tags are used in training and testing, and 91% (88%) with automatically assigned tags.[2]

Table 1 provides summary statistics that describe the amount of fragmentation in the projected dependencies. Contrary to our expectations, the degree of fragmentation is hardly correlated with the increasing distance of the TLs from English: The Italian and German trees are more connected than the Dutch trees. While the growth of fragment size (words/frag) is explained by its correlation with sentence length, the simultaneous decrease in the number of fragments per sentence is astonishing: It means that more edges can be projected to Italian than to the more similar Dutch or German.

We conjecture that this may be due to more literal translations, which in turn allow for better alignments. As it stands, however, these are purely quantitative properties which do not guarantee qualitative equivalents.

3 Parsing with Fragments

Our parser, fMalt, is a variant of the transition-based MaltParser (Nivre et al., 2006). Transition-based parsers construct trees in a stepwise fashion: At each point, the locally optimal parser action (*transition*) t^* is determined on the basis of the current configuration c (previous transitions plus local features):

$$t^* = \arg\max_{t \in T} s(c, t)$$

where T is the set of possible transitions. MaltParser implements an incremental, deterministic parsing algorithm and learns the transition scores s by means of support vector machines (SVMs).

What distinguishes fMalt from the original MaltParser is that it demotes unattached words to serve as mere context rather than headed nodes. It achieves this by eliminating their effect on the margin learned by the SVMs. Since MaltParser scores local decisions, this amounts to suppressing SVM training instances for those words. That is, fragment roots provide context, but no indication of where they are attached themselves. Thus, fMalt can be trained on arbitrarily sparse dependencies. But we do want to exclude uninformative analyses, so we limit the admissible fragmentation to three fragments per sentence and discard sentences that exceed this threshold.[3] Table 2 describes the effect of this filter.

4 Experimental Results

We evaluate fMalt against excerpts from manually annotated treebanks of the three languages.[4] For Dutch and Ger-

[1] As the asymmetry of the IBM translation models only allows for 1–n alignments for a given language pair, we followed standard practice and computed alignments in both directions (SL→TL and TL→SL) which were then intersected.

[2] See also Section 6.

[3] This parameter was fixed after preliminary tests on automatically labeled development data for Dutch. It was then assumed for German and Italian without further tuning. The same is true for various fMalt parameters (cf. Spreyer and Kuhn (2009)).

[4] We replaced the gold POS tags in the test data by the tags assigned automatically by the TreeTagger. This was done in order to

lang.	words/sent	words/frag	frags/sent
nl	9.45	4.31	2.20
de	9.07	4.06	2.23
it	9.85	4.44	2.22

Table 2: Fragmentation in the training samples.

lang.	$\text{baseline}_{\text{unsup}}$	$\text{baseline}_{\text{pos}}$	fMalt
nl	27.63	41.52	**66.58**
de	7.13	39.21	**61.67**
it	50.06	57.72	**66.40**

Table 3: UAS of baselines and fMalt.

man, these are the test sets used in the CoNLL shared tasks (Buchholz and Marsi, 2006): 5,000 words each from the Dutch Alpino Treebank (van der Beek et al., 2002) and the German Tiger Treebank (Brants et al., 2002) For Italian, we use 6,000 words of newspaper text from the Turin University Treebank[5]. The data sets are largely comparable in terms of size and genre (newpaper, out-of-domain). However, a comparison of parser accuracies across languages can only be tentative since the annotation schemes are not immediately compatible. For evaluation purposes, we eliminated some obvious differences,[6] but some errors can still be traced back to such discrepancies (see Section 5).

4.1 Baselines

Along with the unlabeled attachment scores (UAS) of fMalt trained on 100,000 words, Table 3 reports the scores obtained by two heuristic baselines. The first strategy is entirely unsupervised and simply attaches each word to the preceding word (Italian), the following word (Dutch), or the root node (German).

The second baseline attaches words on the basis of their POS tag. The attachment direction for a given tag is estimated from a small set of 10 annotated sentences (excluded from the test set for their evaluation); alternatively, the direction could be provided by a native speaker. If a tag has not been encountered in the training sentences, the direction is assigned by $\text{baseline}_{\text{unsup}}$.

4.2 Discussion

While we do not offer a comparative interpretation of the results here due to the test set heterogeneity described above, we can contrast the intra-language scores and situate our fMalt systems relative to the baselines. As shown in Table 3, all fMalt parsers achieve scores well above their baselines (+8–25%). The baseline performance itself varies immensely: The unsupervised left-attachment strategy for Italian is already relatively successful (50.06%), and–unsurprisingly–the POS-based baseline comes comparatively close to the fMalt system (Δ8.68%). The flat baseline structure built for German proves inadequate (7.13%), but outperforms both left- and right-attachment (not shown). This is probably an artifact of the flat Tiger annotation scheme. Again, the supervised $\text{baseline}_{\text{pos}}$ provides a considerable improvement (+32.08%) over $\text{baseline}_{\text{unsup}}$, but clearly lags behind fMalt (Δ22.46%). $\text{Baseline}_{\text{unsup}}$ for Dutch (right-attachment) suggests an intermediate stage between head-initial and head-final constructions (type or token). Accordingly, $\text{baseline}_{\text{pos}}$ achieves further improvement (+13.89%), but is still outperformed by fMalt by Δ25.06%.

Note that both baselines construct only local dependencies, so their performance also reflects the degree of non-locality in the three languages.

		$\text{baseline}_{\text{unsup}}$	$\text{baseline}_{\text{pos}}$	fMalt
nl	noun	77.44	75.92	**33.46**
	verb	93.03	58.50	**38.45**
	prep	100.00	38.18	**34.44**
	det	27.23	27.23	**7.44**
	adj	31.70	31.29	**27.38**
	adv	70.56	70.83	**59.39**
	comp	98.18	98.15	**63.27**
	other	100.00	98.03	**67.89**
de	noun	97.58	68.77	**35.47**
	verb	57.90	66.16	**34.19**
	prep	98.96	67.96	**47.79**
	det	100.00	35.64	**23.98**
	adj	99.54	30.28	**20.66**
	adv	100.00	61.98	**61.20**
	comp	95.07	54.59	**43.60**
	other	96.53	81.29	**69.38**
it	noun	46.34	46.03	**34.65**
	verb	48.07	47.98	**35.55**
	prep	34.63	35.05	**30.53**
	det	95.52	18.84	**13.29**
	adj	31.36	31.67	**29.19**
	adv	73.38	73.36	**52.59**
	comp	78.30	65.71	**54.06**
	other	69.08	68.97	**68.60**

Table 4: Error rates across word classes.

5 Analysis

Table 4 shows error rates (ER) per word class. The fMalt parsers outperform both baselines across all classes, even those that the baseline is already handling well, such as determiners. This confirms that the parsers indeed learn more than just the simplest attachments. Looking at individual classes, we first notice a comparatively low ER for determiners in all languages. This is especially clear in Dutch (7% ER), and suggests it is no coincidence that the attach-right baseline strategy performs best here. It might be ex-

establish the conditions encountered in the training phase, where no gold standard tags are available.

[5] `http://www.di.unito.it/~tutreeb`

[6] Specifically, we performed *ad hoc* tree transformations to obtain unified analyses of NPs (nominal head rather than DP), PPs (embedded NP rather than flat), coordination (right-branching) and subordination (subordinate clause headed by complementizer rather than verb). In contrast to the transformations reported in Hwa et al. (2005), the changes we make to the trees are mere reformulations of the structure that is already encoded in the original trees. We do not build additional structure.

	nl	de	it
precision	73.00	79.17	34.27
recall	75.21	79.16	14.20

Table 5: Precision and recall for subjects.

data set	en	nl	de	it
original	8.39			
projected		7.47	7.19	5.50
training		12.04	12.58	8.87
application (test)		6.81	8.03	3.96
gold (test)		6.04	7.96	6.14

Table 6: Proportion of subjects at various stages.

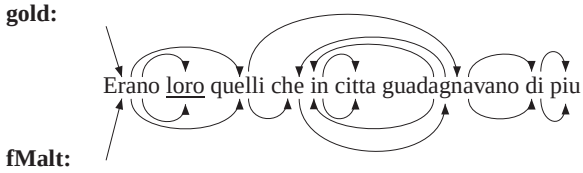

Figure 2: Sentence with a postverbal subject (lit.: 'were they those that in city earned of more', *It was them who earned more in the city*).

	en	de
baseline	91.67	87.13
combined	**91.88**	**88.27**

Table 7: UAS of source language parsers, on automatically assigned POS tags.

plained by an interaction of a predominance of determiners in the test set and their consistent NP-initial realization.

Similarly, adjectives are attached with relative accuracy in all languages because there is little variation in their positioning with respect to the head (left in Dutch and German; right in Italian).

The projection of the subject dependency relation from English to Italian is an interesting test case for the ability to deal with syntactic differences that go along with a (moderate) divergence in the token distribution. As is well-known, Italian is a pro-drop language, i.e., pronominal subjects can be left unrealized. Under perfect alignment, projecting from English to Italian is unproblematic, since the English pronominal subject has no Italian correspondence, so the subject arc is correctly deleted in the projected dependency tree. However, statistical word alignment is known to be error-prone for closed class words such as pronouns. So the English pronoun is occasionally incorrectly aligned with other material.

A further complication comes from the fact that overt Italian subjects can appear in two alternative positions: the standard preverbal position and the sentence-final focus position. Combined with the pro-drop issue, this may give rise to both a relatively high proportion of false positives among the projected subjects, and false negatives of unrecognized Italian subjects. As the precision and recall figures in Table 5 show, the training of the subject dependency relation in Italian is indeed substantially harder than for German and Dutch, even though the basic word order of Italian is more similar to English. Table 6 shows the proportion of subjects among all relations at various stages of the projection procedure. We see that almost three times as many subject edges are lost during projection to Italian as for German or Dutch. Subjects are more frequent in the training sets, where the fragmentation restriction favors shorter sentences. Consequently, we observe overgeneration of subjects by the Dutch and German parsers on the test data (+0.77/0.07%). In contrast, the Italian parser is reluctant to predict subjects (-2.18%).

In Figure 2 fMalt correctly recognizes a subject in non-standard position (*loro*), but we see how two other edges are considered incorrect in the evaluation because the WSJ annotation scheme that fMalt is trained on chooses the relativizer *che* as the head of the relative clause, while it is headed by the verb *guadagnavano* according to the Italian scheme.

6 Impact of Source Parser Quality

So far, we have varied only the target language. But the choice of the source language is at least as important, because it determines the range and quality of resources that can form the basis of projection. In this section, we present experiments with German as an alternative source language, and we investigate the extent to which improvements of the SL parser carry over to the projected TL parser. Like the English parser used in the experiments above, the German parser is a MaltParser, trained on the German Tiger Treebank (Brants et al., 2002). The performance of both parsers is given in Table 7 (labeled 'baseline'). Note that the parsers are trained and tested with POS tags assigned automatically by the TreeTagger, since that is the setting we face when parsing the Europarl source data.

Øvrelid et al. (2009) present a technique for enhancing data-driven dependency parsers with information from large-scale broad-coverage LFG grammars (Kaplan et al., 2004; Forst et al., 2004). They show that for English, the parser can benefit significantly from knowing the dependency structure proposed by the deep grammar. For German, including a variety of grammar-derived morphological, syntactic and semantic features in addition to the dependency structure yields even bigger improvements. The performance of these enhanced parsers is shown in Table 7 (labeled 'combined'). For both languages, the improvement is significant (en: $p<0.01$, de: $p\ll0.01$) according to Dan Bikel's randomized parsing evaluation comparator.[7]

Section 6.1 describes the combined source parsers in more detail. We then present projection experiments based on these source parsers in section 6.2.

[7] `http://www.cis.upenn.edu/~dbikel/software.html#comparator`

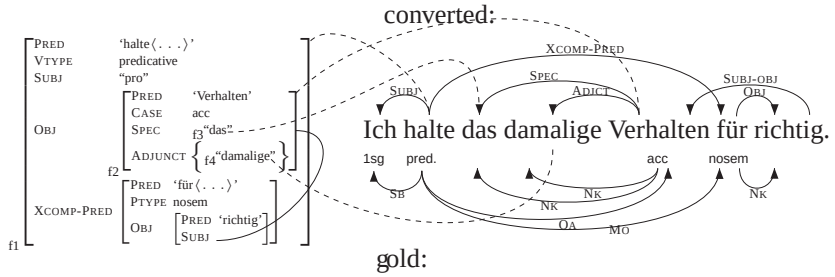

Figure 3: Treebank enrichment with LFG output for German example sentence *Ich halte das damalige Verhalten für richtig* 'I consider the past behavior (to be) correct'.

ID	FORM	POS	FEATS	HEAD	DEPREL	XHEAD	XDEP	XPOS
1	Ich	PPER	pers:1|num:sg|ntype:pron_pers|case:nom	2	SB	2	SUBJ	PPRO
2	halte	VVFIN	mood:indicative|passive:-|vtype:main|tense:pres	0	ROOT	0	ROOT	V
3	das	ART	_	5	NK	5	SPEC	ART
4	damalige	ADJA	atype:attributive|degree:positive	5	NK	5	ADJUNCT	ADJ
5	Verhalten	NN	count:+|num:sg|pers:3|gend:neut|case:acc|def:+|ntype:common	2	OA	7	SUBJ-OBJ	NN
6	für	APPR	psem:nosem	2	MO	2	XCOMP-PRED	PREP
7	richtig	ADJD	atype:predicative|degree:positive|case:acc	6	NK	6	OBJ	ADJ

Table 8: Enhanced treebank version of the German sentence *Ich halte das damalige Verhalten für richtig*.

6.1 Improved SL Parsers: Parser Stacking

We boost the performance of our source parsers by means of a technique of *parser stacking* very similar to the work on combination of dependency parsers in Nivre and McDonald (2008). Parser stacking enables one parser to learn from the output of another parser, in addition to the gold standard treebank annotations. In our case, the data-driven parser is supplemented by the output of a large-scale LFG grammar. In order to include the additional information in the training data, the treebank employed for training of the data-driven baseline parser is parsed with the XLE platform (Crouch et al., 2008), and the output f-structures (shown on the left of Figure 3) are subsequently converted to dependency structures, so that we have two parallel versions of the treebank – one gold standard and one with LFG-annotation (right side of Figure 3, bottom resp. top). We extend the gold standard treebank with the additional information from the corresponding LFG analysis and train the data-driven parser on the enhanced data set. For a detailed description, the reader is referred to Øvrelid et al. (2010).

Treebank representation. Table 8 shows the enhanced treebank version of the example sentence. For each token, the treebank contains information on the word form (FORM), POS tag (POS), as well as the head and dependency relation (HEAD, DEPREL). The added LFG information resides in the FEATS-column and in the additional columns labeled XHEAD and XDEP.

Different annotation schemes. There are interesting differences between the LFG and treebank annotations. Most notably, LFG grammars allow for structure sharing. This can be seen in the f-structure in Figure 3, where the substructure for *Verhalten* 'behavior' is the object of the verb *halte* 'consider' as well as the subject of the predicative adjective *richtig* 'correct'. Since multiple heads cannot be represented in a conventional dependency tree, we resolve shared dependents to the closest head but mark the dependence on another head in a complex label (here, SUBJ-OBJ). Another difference (not evident from the example) arises from the treatment of auxiliary verbs. The treebank annotations always treat the finite verb as the matrix verb, hence the lexical verb in an auxiliary construction is a dependent of the auxiliary. This stands in sharp contrast to the analysis of the grammars, which do not represent the auxiliary explicitly, but rather encode its contribution in the form of features of the main verb. Subjects and certain modifiers are consequently attached to different nodes in the two annotation schemes.

There is of course also a difference in quality, since the grammar output has not been manually corrected. It 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.

Feature model. Finally, 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 standard feature model. We extend the feature model of the baseline parsers using the technique employed in Nivre and McDonald (2008) which allows us to add the predictions of another parser as features for the current parser. In this case we want to add the dependency structure proposed by the LFG grammar as a feature for our data-driven parser. The extended treebank representation (Table 8) readily allows us to refer to the head (XHEAD) and dependency relation (XDEP). Thus, in each parse configuration, we add the proposed dependency relation for the token on top of the stack and for the next input token as features for the parser. We furthermore add a feature which indicates whether there is an arc between these two tokens in the dependency structure (Left, Right, or None).

In order to incorporate further information supplied by the

SL	SL parser	TL: nl
en	baseline	63.33
	combined	63.34
de	baseline	67.49
	combined	**68.20**

Table 9: Impact of SL parse quality on projected Dutch parsers.

	en		de	
	baseline	combined	baseline	combined
noun	37.91	37.61	32.51	**31.63**
verb	39.12	39.06	37.25	37.45
prep	**33.87**	35.45	37.89	36.48
det	7.15	7.02	6.38	6.63
adj	28.51	28.36	26.13	**24.83**
adv	63.65	62.34	52.84	51.57
comp	63.82	62.82	69.09	**67.00**
other	97.93	98.26	58.31	57.28

Table 10: Error rates of Dutch fMalt parsers, projected from various source parsers, across word classes.

LFG grammars we extend the feature models with an additional, static attribute which reads the range of additional linguistic features (FEATS). In addition, the German model refers to the POS tag assigned by the grammar (XPOS).

Parser Accuracy. As mentioned above, the combined parsers produce significantly more accurate analyses. The English parser is improved by 0.21%, the German one by 1.14% UAS (Table 7). In the following, we assess the impact of these improvements on the projected parsers.

6.2 Experiments

For each source language (English and German), we parsed sentences with both the baseline parser and the improved combined parser, and then projected the resulting parse trees to Dutch. Table 9 shows the UAS of Dutch fMalt parsers trained on 50,000 words of the respective projections. Comparing first the SLs, irrespective of the particular parser used, we find that we obtain better results with the German source parsers than with the English ones, despite the fact that the latter achieves a substantially higher attachment score (cf. Table 7). This apparent discrepancy can be explained by the relation of SL and TL translations, which may simply be closer between German and Dutch. In addition, it is well known that German is generally tough to parse (Kübler et al., 2006), but error-prone attachment decisions may be resolved or eliminated in the Dutch translations. We conclude from our results that "closeness" of SL and TL translations is likely to outweigh potential performance deficits of the particular SL parser.

However, when the SL is fixed we see that performance improvements of the source parser can carry over to the projected TL parser. While the difference between fMalt projected from the baseline vs. the combined parser is not significant[8] for English (Δ0.01% UAS), we observe a significant ($p<0.05$) difference of Δ0.71% for German.

Table 10 breaks the results down by word class. The overall trends observed in Table 9 are confirmed here, namely that German appears to be a more suitable source for projection to Dutch than English, and that better source parsers give rise to more accurate projected parsers, for most word classes. Especially for the parser projected from German, we find significant improvements for nouns, adjectives and complementizers, and notable (albeit not significant) changes for prepositions and adverbials. In fact, these are the same types of improvements revealed in an error analysis of the parser stacking for German (Øvrelid et al., 2010): Whereas we observe a general improvement for argument relations, such as subjects and objects in both combined parsers, we find that the analysis of adverbials to a larger extent improves for German. The modifier relation MO which is employed largely for prepositional phrases at the sentence level, as well as some adjectives and adverbials is one of the relations for which parser performance improves the most in the German combined parser.

7 Related Work

Annotation projection has been applied to many different NLP tasks. On the word or phrase level, these include morphological analysis, part-of-speech tagging and NP-bracketing (Yarowsky et al., 2001), temporal analysis (Spreyer and Frank, 2008), semantic lexicon induction (Padó and Lapata, 2005), or semantic role labeling (Padó and Lapata, 2006). In the word level tasks, labels can technically be introduced in isolation, without reference to the rest of the annotation. This means that unreliable data points can be discarded by aggressive filters, and conversely, that gaps in the projection (e.g., due to missing alignments) do not affect the wellformedness of other projected material in the same sentence. The annotation of NP brackets, temporal expressions or semantic roles, on the other hand, is more constrained in that the target structures may encompass multiple words, and the plain projections are typically completed to form contiguous spans.

On the sentence level, Hwa et al. (2005) were the first to project dependency trees from English to Spanish and Chinese. They identify unreliable target parses (as a whole) on the basis of the number of unaligned or over-aligned words. In addition, they manipulate the trees by inserting extra nodes to accommodate for non-isomorphic sentences. Systematic non-parallelisms between source and target language are then addressed by hand-crafted rules in

[8] A cross-validation scheme is not applicable with a monolingual test set. Further complication arises from the fact that the underlying source language parsers differ not only 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 over the results of training on 10 random samples from the respective training data. We report the mean of these results.

a post-projection step. These rules correct projected dependencies and introduce new ones so as to build a connected tree. The manually designed transformations account for an enormous increase in the unlabeled f-score of the direct projections, from 33.9 to 65.7 for Spanish and from 26.3 to 52.4 for Chinese. But they need to be designed anew for every target language, which is time-consuming and, more importantly, requires knowledge of that language. By contrast, our approach of training directly on fragmented dependency trees allows us to remain agnostic about dependencies that cannot be transfered reliably from the source language. While fMalt itself is not capable of inducing structure that has never been observed in the training examples, it should be straightforward to combine fMalt with machine learning schemes of various degrees of supervision that are suited to discover additional language-specific knowledge automatically, or learn it from minimal amounts of annotated data.

8 Conclusion

We have shown that training on tree fragments is an inexpensive and effective way to obtain parsers for a diverse range of languages. Analysis of the annotations projected from English reveals that the distance between SL and TL has little influence on fragmentation. Of course, this depends crucially on the word alignment, which ideally encapsulates language-specific traits such as word order differences, allowing them to be recovered under projection.
We have also demonstrated that improvements of the source language parser, e.g., by means of parser stacking, directly carry over to the projected target language parser. This means that projection approaches can benefit immediately from current advances in monolingual dependency parsing. Although the projected parsers cannot close the gap to state-of-the-art supervised treebank parsers in terms of accuracy, they clearly outperform the POS-based baseline. In other words, while parser projection cannot fully replace manual annotation, it does provide a cheap and efficient basis for manual or (semi-)automatic correction. This, in turn, promotes substantial speed-ups for the creation of large-scale annotated resources.

Acknowledgments

The work reported in this paper was in part 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).

References

Thorsten Brants, Wolfgang Lezius, Oliver Plaehn, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of CoNLL-X*, pages 149–164, New York City, June.

Dick Crouch, Mary Dalrymple, Ron Kaplan, Tracy King, John Maxwell, and Paula Newman. 2008. XLE documentation. Palo Alto Research Center (PARC), available Online.

Martin Forst, Berthold Crysmann, Frederik Fouvry, Silvia Hansen-Schirra, and Valia Kordoni. 2004. Towards a dependency-based gold standard for German parsers – The TiGer Dependency Bank. In *Proceedings of the Workshop on Linguistically Interpreted Corpora*.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3):311–325.

Ron Kaplan, Stefan Riezler, Tracy King, John Maxwell, Alexander Vasserman, and Richard Crouch. 2004. Speed and Accuracy in Shallow and Deep Stochastic Parsing. In *Proceedings of HLT-NAACL 2004*.

Sandra Kübler, Erhard W. Hinrichs, and Wolfgang Maier. 2006. Is it really that difficult to parse German? In *Proceedings of EMNLP 2006*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of 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*, pages 950–958, Columbus, Ohio, June.

Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiğit, and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of CoNLL-X*, pages 221–225.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Lilja Øvrelid, Jonas Kuhn, and Kathrin Spreyer. 2009. Improving data-driven dependency parsing using large-scale LFG grammars. In *Proceedings of the Annual Meeting for the Association for Computational Linguistics (ACL) (Short Paper)*.

Lilja Øvrelid, Jonas Kuhn, and Kathrin Spreyer. 2010. Cross-framework parser stacking for data-driven dependency parsing. To appear in TAL 2010 special issue on Machine Learning for NLP 50(3), eds. Isabelle Tellier and Mark Steedman.

Sebastian Padó and Mirella Lapata. 2005. Cross-lingual Bootstrapping for Semantic Lexicons: The Case of FrameNet. In *Proceedings of AAAI 2005*, pages 1087–1092, Pittsburgh, PA.

Sebastian Padó and Mirella Lapata. 2006. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of COLING/ACL 2006*, Sydney, Australia.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, England.

Kathrin Spreyer and Anette Frank. 2008. Projection-based

acquisition of a temporal labeller. In *Proceedings of IJCNLP 2008*, Hyderabad, India, January.

Kathrin Spreyer and Jonas Kuhn. 2009. Data-driven dependency parsing of new languages using incomplete and noisy training data. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 12–20, Boulder, CO, June. Association for Computational Linguistics.

Leonoor van der Beek, Gosse Bouma, Robert Malouf, and Gertjan van Noord. 2002. The Alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN)*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001*.