Improving the Cross-Lingual Projection of Syntactic Dependencies

Jörg Tiedemann
Uppsala University
Department of Linguistics and Philology
firstname.lastname@lingfil.uu.se

Abstract

This paper presents several modifications of the standard annotation projection algorithm for syntactic structures in cross-lingual dependency parsing. Our approach reduces projection noise and includes efficient data sub-set selection techniques that have a substantial impact on parser performance in terms of labeled attachment scores. We test our techniques on data from the Universal Dependency Treebank and demonstrate the improvements on a number of language pairs. We also look at treebank translation including syntax-based models and data combination techniques that push the performance even further. We achieve absolute improvements of up to over seven points in labeled attachment scores pushing the state-of-the art in cross-lingual dependency parsing for all language pairs tested in our experiments.

1 Introduction

State-of-the art dependency parsing is mainly based on annotated data and supervised learning techniques. This, however, restricts the use of parsing technology to a few languages for which sufficient amounts of training data is available. Fully unsupervised techniques still fall far behind in their performance and cannot produce labels that are necessary for many downstream applications. Cross-lingual learning techniques have, therefore, been proposed as a quick solution to bootstrap tools for otherwise unsupported languages. There are basically two strategies that can be found in the literature: annotation projection and model transfer.

Model transfer has attracted a lot of interest recently due to the availability of cross-lingually harmonized annotation (Petrov et al., 2012) that makes it possible to use universal features across languages. The most straightforward technique is to train delexicalized parsers that heavily rely on universal POS tags. This simple technique has shown some success for closely related languages (McDonald et al., 2013). Several improvements can be achieved by using multiple source languages (McDonald et al., 2011; Naseem et al., 2012) and additional cross-lingual features that can be used to transfer models to a new language such as cross-lingual word clusters (Täckström et al., 2012) or word-typology information (Täckström et al., 2013).

Annotation projection has already a long tradition in NLP. Initially proposed for tasks like POS tagging (Yarowsky et al., 2001), the seminal work for annotation projection in dependency parsing is presented by Hwa et al. (2005). The general idea is to make use of parallel corpora and automatic word alignment to transfer information from the source language to a target language translation that can then be used for training parsers. In most cases, treebanks are not taken from parallel corpora and, therefore, one has to rely on automatic annotation of the source language part of another (usually unrelated) bitext. Together with the noise in automatic word alignment, these steps are bottlenecks in the projection strategy. Hwa et al. (2005) propose the basic projection heuristics (which they call the direct correspondence assumption algorithm or DCA for short) that can handle various types of word alignments. In this paper we revisit this algorithm and include a systematic comparison of projection heuristics together with various modifications and data-set selection techniques. We can show that these methods lead to significant improvements for all languages tested in our experiments.

Finally, we also look at the recently proposed treebank translation approach (Tiedemann et al., 2014), which can be used as an alternative to annotation projection on existing parallel data sets. Automatic translation has the advantage that we

can use the manually verified annotation of the source language treebank instead of noisy machine-annotated parallel data and also the given word alignment, which is an integral part of the translation model. We present additional improvements when using our modifications of the projection algorithm and also show a positive effect when combining projected data from parallel corpora and machine translated treebanks.

2 Projection Using Parallel Corpora

Our first batch of experiments is based on the projection of syntactic information using existing parallel corpora. The basic setup is as follows:

1. Parse the source side of the parallel corpus with a parser trained on the source language treebank.
2. Project the syntactic information (including POS labels) to the target side of the parallel corpus using word alignment links and the direct correspondence assumption.
3. Train a parser on the projected data and evaluate its performance on the test sets of the universal treebank for the target language.

Word alignments are produced using IBM model 4 as implemented in GIZA++ (Och and Ney, 2003) trained in the typical pipeline as it is common in statistical machine translation using the Moses toolbox (Koehn et al., 2007). The asymmetric alignments are symmetrized with the intersection and the grow-diag-final-and heuristics (Koehn et al., 2003). We use the latter for the basic annotation projection presented in the next section.

For evaluation, we use the test sets provided by the Universal Dependency Treebank (UDT) version 1 (McDonald et al., 2013). The harmonized annotation makes it possible to perform a fair evaluation across languages including labeled attachment scores, which we use as our essential evaluation metric. Note that all scores include attachments of punctuation which makes our results directly comparable to the results presented in the related literature (Tiedemann, 2014).

2.1 Baseline

For our experiments, we use 40,000 sentences from Europarl (Koehn, 2005) for each language pair following the basic setup of Tiedemann (2014). The baseline model applies the projection heuristics as presented by Hwa et al. (2005):

one-to-one: For one-to-one alignments between the source words s_i and s_j and the target words t_x and t_y: Copy the relation $R(s_i, s_j)$ to $R(t_x, t_y)$.

unaligned source: Add dummy nodes in the target language that take all incoming and outgoing arcs of the unaligned source language word.

one-to-many: Add a dummy node in the target sentence and attach the aligned target words to this node (using a dummy label as well) and remove the original word alignments. Align the newly created dummy word with the corresponding source language word.

many-to-one: Retain only the link between the target language word and the source language word that is the highest up in the source language tree and delete all other links.

many-to-many: Perform the rule for one-to-many alignments first and then perform the rule for many-to-one alignments.

unaligned target: Remove all unaligned target words.

These heuristics ensure that the projected structures are proper trees and that we can train dependency parsers that are capable of handling non-projective structures without modification. Note that POS tags are also projected along the remaining word alignments and that some words obtain dummy tags if there is no relation to a source language token that could be used for projection.

In all our experiments, we apply MaltParser (Nivre et al., 2006) to train transition-based dependency parsers and we optimize feature models and learning parameters using MaltOptimizer (Ballesteros and Nivre, 2012). The parameters and feature models for the cross-lingual models are directly copied from the source language model in order to apply a realistic scenario for which no tuning data for the target language would be available. Table 1 lists the results in terms of labeled attachment scores of our baseline models for all language pairs in the test set. Rows correspond to each source language and columns represent the target language used for testing. Note that we restrict all our experiments to the languages for which the same kind of parallel data is available in Europarl.

The baseline scores are mainly in the range of 50-60% LAS with closely related languages (like French and Spanish) performing slightly better. This is on par with previously reported scores.

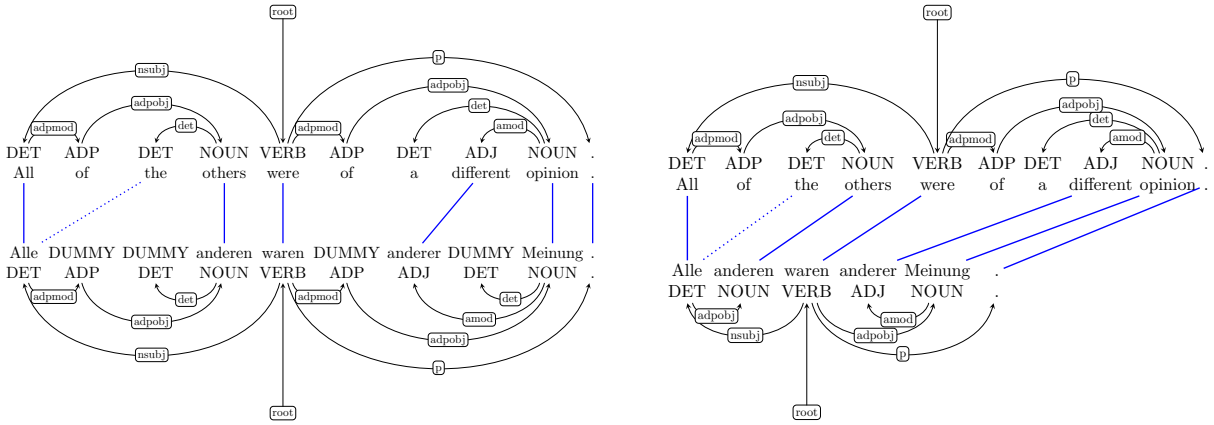


Figure 1: Removing unnecessary dummy nodes (right image) from standard DCA-based annotation projection (left image).

	DE	EN	ES	FR	SV
DE	(72.13)	48.81	56.76	58.52	60.33
EN	55.78	(87.50)	60.27	61.86	61.41
ES	52.94	47.74	(78.54)	65.12	60.97
FR	53.08	50.55	64.41	(77.51)	57.60
SV	55.12	48.76	60.76	61.60	(81.28)

Table 1: Baseline performance in LAS of a DCA-based annotation projection with 40,000 sentences (models trained on the original treebanks in grey).

2.2 Removing Unnecessary Dummy Nodes

A consequence of the projection heuristics is the appearance of dummy nodes and dummy labels. This may have a significantly negative impact on the performance of the model that is trained on this kind of data. Tiedemann et al. (2014) already discuss this problem and they propose an alternative projection algorithm, which, however, is not very successful in their experiments. In this work, we propose some different techniques that can be used to reduce or even remove all dummies from the data and we can show that these techniques are very effective.

The first method is similar to the approach presented by Tiedemann (2014). Arcs that run over dummy nodes that connect to a single daughter node only can simply be collapsed without any changes in the remaining structure. Figure 1 illustrates an example with two such unary dummy nodes that can be removed. The main difficulty with this method is to decide on the label for the arc that corresponds to the two collapsed ones. In some cases, one of the arcs is labeled as dummy as well and could, therefore, easily be ignored. This is not the case in our example and we decided to always use the label of the outgoing arc as illustrated in Figure 1.

In addition to collapsing unary dummy nodes, we can also ignore dummy nodes that are leaves of the dependency tree. Here, we assume that these nodes do not contribute much to the information projected from the source and rather confuse the learning algorithm. Figure 1 illustrates this procedure as well with two dummy determiners removed from the projected tree.

	DE	EN	ES	FR	SV
DE	–	**48.87**$^{(0.06)}$	**57.52**$^{(0.76)}$	**58.83**$^{(0.31)}$	**61.62**$^{(1.29)}$
EN	**56.64**$^{(0.86)}$	–	60.12$^{-0.15}$	**62.13**$^{(0.27)}$	**62.89**$^{(1.48)}$
ES	**53.77**$^{(0.83)}$	47.24$^{-0.50}$	–	**66.00**$^{(0.88)}$	60.65$^{-0.32}$
FR	**53.44**$^{(0.36)}$	49.69$^{-0.86}$	**64.69**$^{(0.28)}$	–	**59.16**$^{(1.56)}$
SV	**55.62**$^{(0.50)}$	**49.23**$^{(0.47)}$	60.47$^{-0.29}$	**61.86**$^{(0.26)}$	–

Table 2: Collapsing arcs over unary dummy nodes and removing dummy leaves (difference to baseline in superscript).

Table 2 summarizes the LAS scores after transforming our data sets in the way described above. We can see that this rather trivial change has positive effects on most models. In some cases there are substantial gains in LAS. However, we can also observe slight drops in performance for a few language pairs, which we should investigate in more details in future work.

2.3 Alternative Treatment of MWU's

Another consequence of the DCA algorithm is the insertion of dummy nodes which serve as heads of multi-word units that are aligned to single words in the source language. The left tree in Figure 2 illustrates this behavior with a dummy noun that

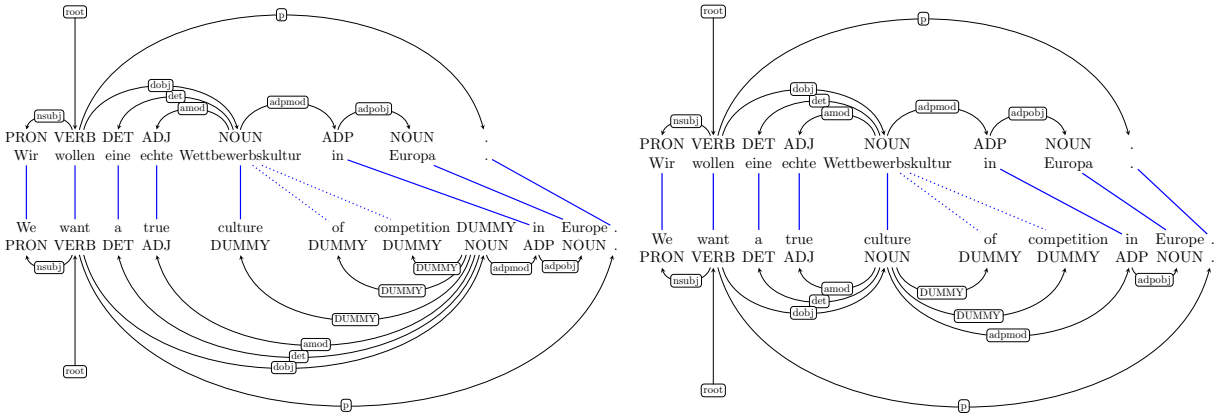


Figure 2: Projecting from German to English using the default DCA algorithm (left image) and using the new treatment for one-to-many word alignments (right image). Dotted lines are links from the grow-diag-final-and symmetrization heuristics and solid lines refer to links in the intersection of word alignments.

covers the noun phrase "culture of competition" which is aligned to "Wettbewerbskultur" in German. However, in contrast to the original setup of the DCA-based annotation projection, we have several word alignments at our disposal based on different symmetrization heuristics. The idea in our approach is now to make use of high-precision links to determine the head connection between source and target and to use other links to attach the remaining tokens. Figure 2 illustrates the procedure with the given example. In the figure, solid lines refer to high-precision links coming from the intersection of directional word alignments whereas dotted lines refer to additional links coming from the grow-diag-final-and heuristics that gives higher coverage. As we can see in the figure, "culture" is then chosen as the head of the multi-word unit and the other tokens in the NP are attached as dummy relations. This treatment is certainly not ideal but lacking more information we have at least eliminated yet another dummy node in our projected tree in a reasonable way.

The results of this procedure are summarized in Table 3. We can see that the new treatment of one-to-many links has again an overall positive effect on parsing performance with modest gains in most cases. It should be noted that the head selection heuristics is by far not perfect and that not all multi-word units can be resolved in this way. In many cases, none of the links is part of the intersection of links and, consequently, the projection algorithm has to fall back to the standard treatment with additional dummy nodes.

	DE	EN	ES	FR	SV
DE	–	**49.62**$^{(0.81)}$	**57.54**$^{(0.78)}$	**59.60**$^{(1.08)}$	**61.80**$^{(1.47)}$
EN	56.47$^{(0.69)}$	–	**60.57**$^{(0.30)}$	**62.71**$^{(0.85)}$	**62.94**$^{(1.53)}$
ES	**53.94**$^{(1.00)}$	**48.02**$^{(0.28)}$	–	65.74$^{(0.62)}$	**61.33**$^{(0.36)}$
FR	53.36$^{(0.28)}$	50.22$^{-0.33}$	64.54$^{(0.13)}$	–	**59.27**$^{(1.67)}$
SV	**56.34**$^{(1.22)}$	**49.30**$^{(0.54)}$	60.66$^{-0.10}$	**62.56**$^{(0.96)}$	–

Table 3: Using the intersection of word alignments to resolve one-to-many links without creating dummy head nodes. Bold numbers are also better than Table 2.

2.4 Data Sub-Set Selection

Yet another possibility for improvements is data selection or instance weighting. Here, we opt for subset selection techniques based on simple heuristic filters, which prove to be very effective for our task. The first idea is to simply discard any projected tree that includes dummy nodes. Our assumption is that such dummy nodes have a negative influence on the learning algorithm but also that sentence pairs, which require complex projection heuristics due to difficult word alignments are in general less suited to be used for annotation projection.

	DE	EN	ES	FR	SV
DE	–	**50.05**$^{(1.24)}$	**58.30**$^{(1.54)}$	**59.67**$^{(1.15)}$	**62.33**$^{(2.00)}$
EN	**58.33**$^{(2.55)}$	–	**61.01**$^{(0.74)}$	**63.34**$^{(1.48)}$	**63.70**$^{(2.29)}$
ES	**55.46**$^{(2.52)}$	**48.05**$^{(0.31)}$	–	65.90$^{(0.78)}$	**61.44**$^{(0.47)}$
FR	**54.39**$^{(1.31)}$	**50.69**$^{(0.14)}$	**65.08**$^{(0.67)}$	–	**60.23**$^{(2.63)}$
SV	**57.84**$^{(2.72)}$	**50.42**$^{(1.66)}$	**60.86**$^{(0.10)}$	62.47$^{(0.87)}$	–

Table 4: Discarding all projected trees that include dummy nodes (bold numbers are also better than Tables 2 and 3).

Table 4 shows the results when applying this simple filter on the data set of 40,000 projected sentences for each language pair. We can see that we obtain some significant improvements over the previous projection even though we reduce the training data substantially. To quantify this reduction, Table 5 lists the sizes of the remaining data sets we obtain. In several cases, the data is reduced to less than 10% of the original set but still performs as well or even better than the full data set of projected trees, which is quite remarkable. Note that all scores are also better than the baseline models.

	DE	EN	ES	FR	SV
DE	–	3778	3069	2557	7966
EN	6166	–	5010	3755	8169
ES	4114	5127	–	4332	4814
FR	5773	6917	7552	–	7104
SV	4661	3198	2484	1671	–

Table 5: Successfully projected trees out of 40,000 sentences when discarding trees with dummy nodes.

In order to perform a fair comparison, we ran another experiment with additional sentences coming from the same parallel corpus that fill up the projected training data to the same size of 40,000 trees as it is used in the other experiments. Table 6 lists the final results after training parser models on these extended data sets. We can see that we obtain yet another significant improvement and the best results for our task so far in almost all cases. Some scores are slightly below the performance of the reduced data set which is a bit surprising.

	DE	EN	ES	FR	SV
DE	–	**50.45**$^{(1.64)}$	**58.65**$^{(1.89)}$	**59.77**$^{(1.25)}$	**62.78**$^{(2.45)}$
EN	58.05$^{(2.27)}$	–	**60.77**$^{(0.50)}$	**64.71**$^{(2.85)}$	**64.34**$^{(2.93)}$
ES	**56.14**$^{(3.20)}$	**48.39**$^{(0.65)}$	–	65.91$^{(0.79)}$	**61.52**$^{(0.55)}$
FR	**55.47**$^{(2.39)}$	**51.15**$^{(0.60)}$	**65.27**$^{(0.86)}$	–	**59.99**$^{(2.39)}$
SV	**57.91**$^{(2.79)}$	**50.10**$^{(1.34)}$	**61.33**$^{(0.57)}$	**62.78**$^{(1.18)}$	–

Table 6: The same setting as in Table 4 but projecting the same number of sentences as in all other experiments (40,000) (bold numbers are higher than all previous settings).

Finally, we also define yet another simple filter that removes all trees that include any kind of dummy relation. Using this filter together with the one above dramatically reduces the size of the data and the scores obtained when training on the projected trees that remain from the original 40,000 sentences is not worthwhile to show here. However, filling up the data with additional sentences pushes the performance yet another step and the final scores are shown in Table 7. For some reason, French as a target language was not very successful with this strategy but in most other cases we can see considerable improvements over the previously noted top scores.

	DE	EN	ES	FR	SV
DE	–	**50.72**$^{(1.91)}$	**58.82**$^{(2.06)}$	59.37$^{(0.85)}$	62.78$^{(2.45)}$
EN	**59.34**$^{(3.56)}$	–	60.72$^{(0.45)}$	64.01$^{(2.15)}$	**64.52**$^{(3.11)}$
ES	**56.29**$^{(3.35)}$	**49.05**$^{(1.31)}$	–	64.62$^{-0.50}$	**62.28**$^{(1.31)}$
FR	**56.07**$^{(2.99)}$	**51.25**$^{(0.70)}$	**65.58**$^{(1.17)}$	–	**60.36**$^{(2.76)}$
SV	**58.04**$^{(2.92)}$	**50.55**$^{(1.79)}$	60.11$^{-0.65}$	61.35$^{-0.25}$	–

Table 7: Discarding all trees that include dummy nodes or dummy labels on any dependency relations but still projecting 40,000 sentences (bold numbers are higher than any previous setting).

3 Translated Treebanks

Treebank translation has been proposed by Tiedemann et al. (2014). In this paper, we would like to explore the impact of our modifications of the projection algorithm on that approach as well. For this, we use the training sets of the Universal Dependency Treebank and translate them with standard SMT models to the target languages we would like to test. Our setup is very generic and uses the Moses toolbox for training, tuning and decoding. The translation models are trained on the entire Europarl corpus version 7 without language-pair-specific optimization. For tuning we use MERT (Och, 2003) and the newstest 2011 data provided by the annual workshop on statistical machine translation.[1] The language model is a standard 5-gram model and is based on a combination of Europarl and News data provided from the same source. We apply modified Kneser-Ney smoothing without pruning, applying KenLM tools (Heafield et al., 2013) for estimating the LM parameters.

3.1 Phrase-based SMT

Our baseline system is a standard phrase-based model and we use the standard DCA projection algorithm as proposed by Hwa et al. (2005). The results are shown in Table 8.

With this, we can confirm the findings of Tiedemann (2014) that the translation approach has some

[1] http://www.statmt.org/wmt14. For Swedish we use a sample from the OpenSubtitles2012 corpus (Tiedemann, 2012).

	DE	EN	ES	FR	SV
DE	–	**53.36**$^{(4.55)}$	54.72$^{-2.04}$	58.07$^{-0.45}$	59.84$^{-0.49}$
EN	53.09$^{-2.69}$	–	**60.81**$^{(0.54)}$	**64.23**$^{(2.37)}$	**63.43**$^{(2.02)}$
ES	50.54$^{-2.40}$	**50.39**$^{(2.65)}$	–	**66.10**$^{(0.98)}$	60.56$^{-0.41}$
FR	49.89$^{-3.19}$	**53.65**$^{(3.10)}$	**65.05**$^{(0.64)}$	–	**58.38**$^{(0.78)}$
SV	53.83$^{-1.29}$	**50.93**$^{(2.17)}$	60.61$^{-0.15}$	60.46$^{-1.14}$	–

Table 8: Treebank translation with DCA-based projection (compared to the projection of parallel data from Table 1).

advantages over the projection of automatically annotated parallel corpora. For some language pairs, the labeled attachment scores are significantly above the projection results even though the parsers are trained on much smaller data sets (the treebanks are typically much smaller than 40,000 sentences for most language pairs). Very striking is also the outcome for German as a target language, which seems to be the hardest language to translate to in this data set.

In the next experiment we apply the same modifications of the projection algorithm as presented in Section 2.2. Once again, we can see that we obtain considerable improvements for most language pairs, which nicely re-assures the general utility of these techniques (see Table 8).

	DE	EN	ES	FR	SV
DE	–	**54.69**$^{(1.33)}$	**56.72**$^{(2.00)}$	57.63$^{-0.44}$	**60.07**$^{(0.23)}$
EN	**53.50**$^{(0.41)}$	–	**61.39**$^{(0.58)}$	**64.63**$^{(0.40)}$	**63.85**$^{(0.42)}$
ES	50.33$^{-0.21}$	49.90$^{-0.49}$	–	**66.37**$^{(0.27)}$	59.96$^{-0.60}$
FR	**51.81**$^{(1.92)}$	**54.85**$^{(1.20)}$	**66.32**$^{(1.27)}$	–	**59.34**$^{(0.96)}$
SV	**53.90**$^{(0.07)}$	**51.18**$^{(0.25)}$	**60.99**$^{(0.38)}$	**61.01**$^{(0.55)}$	–

Table 9: Collapsing relations over unary dummy nodes and removing dummy leave nodes (same approach as in Section 2.2; improvements over Table 8 in superscript)

Unfortunately, it is not possible to straightforwardly test the alternative treatment of multi-word-units presented in Section 2.3 as we do not have alternative word alignments readily available from the translation model. Certainly, additional alignments could be produces but for this, we would need to concatenate the translated treebanks with larger parallel corpora to obtain reasonable statistics for unsupervised word alignment, which still might not work very well. In our current experiments we, therefore, excluded this setup and may return to this idea in future work.

Furthermore, we do not include results with data selection techniques that we discussed in Section 2.4. This strategy is not very successful in the translation-based setup and the reason for this is that the data size drops substantially (having small treebanks to start with already) which causes significant drops in parsing performance.

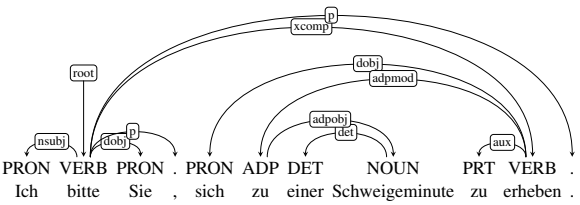
3.2 Syntax-Based SMT

Previous research focused on phrase-based translation models and the projection through word alignments as described in the previous sections. In this paper, we also look at syntax-based SMT, which intuitively provides a better fit for syntactic annotation projection. The main motivation for this is the clear connection between syntax-based translation and syntactic annotation projection.

Syntax-based MT models supported by Moses are based on synchronous context-free grammars which are induced from aligned parallel data. Several modes are available. In our case, we are mostly interested in the tree-to-string models that require syntactic parse trees on the source language side (which we would like to project). Our assumption is that the structural relations that are induced from the parallel corpus with a fixed given source-side analysis improve the projection of syntactic relations when used in combination with syntax-based translation.

In order to make it possible to use dependency information in the framework of synchronous CFGs we convert projective dependency trees to the phrase structures required for training tree-to-string models with Moses. Figure 3 shows an example of an automatically parsed German sentence from Europarl and its conversion. We use the yield of each word to define a span over the sentence which forms a constituent with the label taken from the relation of that word to its head. Certainly, dependency trees using this conversion approach are not optimal for syntax-based SMT as they are usually very flat and do not provide the deep hierarchical structures that are common in phrase-structure trees. However, we still believe that valuable information can be pushed into the model in this way that may be beneficial for projecting dependency relations. Note that we use part-of-speech tags as additional pre-terminal nodes to enrich the information given to the system. The entire procedure in our approach is then as follows:

- We tag the source side of a parallel corpus with a POS tagger trained on the UDT training




```
<tree label="ROOT">
 <tree label="nsubj"><tree label="PRON">Ich</tree></tree>
 <tree label="VERB">bitte</tree>
 <tree label="dobj"><tree label="PRON">Sie</tree></tree>
 <tree label="p"> <tree label=".">,</tree> </tree>
 <tree label="xcomp">
  <tree label="dobj"><tree label="PRON">sich</tree></tree>
  <tree label="adpmod">
   <tree label="ADP">zu</tree>
   <tree label="adpobj">
    <tree label="det"><tree label="DET">einer</tree></tree>
    <tree label="NOUN">Schweigeminute</tree>
   </tree>
  </tree>
  <tree label="aux"><tree label="PRT">zu</tree></tree>
  <tree label="VERB">erheben</tree>
 </tree>
 <tree label="p"><tree label=".">.</tree></tree>
</tree>
```

Figure 3: A dependency tree taken from the automatically annotated parallel data and its conversion to a nested phrase-structure tree in Moses format.

data using HunPos (Halácsy et al., 2007).

- We parse the tagged corpus using a MaltParser model trained on the UDT with a feature model optimized with MaltOptimizer (Ballesteros and Nivre, 2012).
- We projectivize all trees using MaltParser and convert to nested tree annotations.
- We extract synchronous rule tables from the word aligned bitext with source side syntax and score rules using Good Turing discounting. We do not use any size limit for replacing sub-phrases with non-terminals at the source side and restrict the number of non-terminals on the right-hand side of extracted rules to three. Furthermore, we allow consecutive non-terminals on the source side to increase coverage, which is not allowed in the default settings of the hierarchical rule extractor in Moses.
- We tune the model using MERT and the same data sets as before.
- Finally, we parse the training data of the UDT in the source language and translate it to the target language using the tree-to-string model created above.

Similar to the previous section, we then test the performance of our models on the target language test sets from the UDT. Table 10 lists the results in terms of labeled attachment scores.

	DE	EN	ES	FR	SV
DE	–	**54.72**$^{(5.91)}$	**59.87**$^{(3.11)}$	**59.77**$^{(1.25)}$	**63.15**$^{(2.82)}$
EN	**56.56**$^{(0.78)}$	–	**62.29**$^{(2.02)}$	**64.79**$^{(2.93)}$	**63.90**$^{(2.49)}$
ES	***52.40***$^{-0.54}$	**51.39**$^{(3.65)}$	–	65.48$^{(0.36)}$	**61.26**$^{(0.29)}$
FR	***52.56***$^{-0.52}$	**55.17**$^{(4.62)}$	**65.29**$^{(0.88)}$	–	**58.42**$^{(0.82)}$
SV	**55.48**$^{(0.36)}$	50.80$^{(2.04)}$	**61.34**$^{(0.58)}$	***60.52***$^{-1.08}$	–

Table 10: Annotation projection using tree-to-string models for translating treebanks (differences in LAS scores to the projection baseline are in superscript numbers). Results in bold are better than the phrase-based translation (Table 8). Scores in italics are worse than the annotation projection baseline (Table 1).

The results of the syntax-based translation projection are quite impressive. Almost all cases outperform the phrase-based MT approach which shows the potentials of these models for syntactic annotation projection. Furthermore, only three cases are below the annotation projection baseline and for the majority of language pairs we can observe a substantial improvement of up to 5.91 points in LAS compared to that baseline. It is difficult to say why the approach did not work as well for translating Spanish and French to German and Swedish to French but this may be related to specific properties of the treebanks involved and the domain mismatch with the data used for SMT training. Note that phrase-based models performed even worse for these language pairs and that only two other cases are slightly below the phrase-based translation projection whereas other language pairs obtain increased LAS's of several points (see, for example, German-Spanish and German-Swedish) compared to phrase-based SMT.

	DE	EN	ES	FR	SV
DE	–	**54.89**$^{(0.17)}$	**60.11**$^{(0.24)}$	**60.06**$^{(0.29)}$	**63.82**$^{(0.67)}$
EN	56.45$^{-0.11}$	–	**62.57**$^{(0.28)}$	**64.95**$^{(0.16)}$	63.72$^{-0.18}$
ES	**52.90**$^{(0.50)}$	**51.80**$^{(0.41)}$	–	**65.86**$^{(0.38)}$	60.24$^{-1.02}$
FR	**55.03**$^{(2.47)}$	**56.09**$^{(0.92)}$	**66.00**$^{(0.71)}$	–	**59.29**$^{(0.87)}$
SV	**55.70**$^{(0.22)}$	**51.18**$^{(0.38)}$	**61.64**$^{(0.30)}$	**60.91**$^{(0.39)}$	–

Table 11: Treating dummy nodes as described in Section 2.2.; improvements over Table 10)

Finally, we can use the same techniques for removing dummy nodes as described in Section 2.2. The results are shown in Table 11. Again, we can see consistent improvements in LAS with only a few exceptions.

4 Discussions

One of the questions that we have is whether there is a correlation between translation quality and the performance of the cross-lingual parsers based on translated treebanks. As an approximation for treebank translation quality we computed BLEU scores over well-established MT test sets from the WMT shared task, in our case newstest 2012.[2]

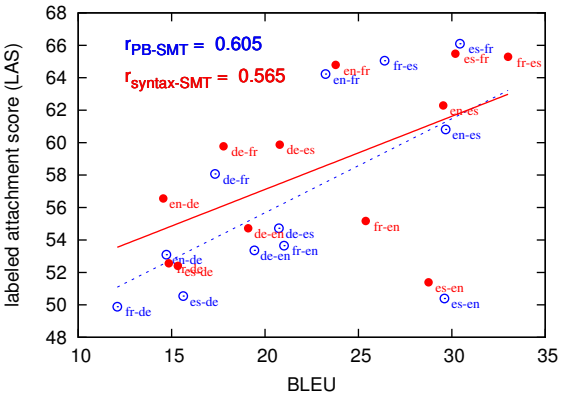


Figure 4: Correlation between BLEU scores and cross-lingual parsing accuracy.

Figure 4 illustrates the correlation between BLEU scores obtained on newstest data and LAS's of the corresponding cross-lingual parsers. First of all, we can see that the MT performance of phrase-based and syntax-based models is quite comparable with some noticeable exceptions in which syntax-based SMT is significantly better (French-English and French-Spanish, which is rather surprising). However, looking at most language pairs we can see that the increased parsing performance does not seem to be due to improvements in translation but rather due to the better fit of these models for syntactic annotation projection (see German, for example). Nevertheless, we can observe a correlation between BLEU scores and LAS within a class of models with one notable outlier, Spanish-English. This correlation may be explained by the fact that language relation is a crucial factor for both tasks, machine translation and annotation projection, with French and Spanish as the top-performing language pair in our experiments.

Another interesting question is whether the different data sets can successfully be combined. In order to test this possibility, we conducted a final experiment in which we concatenated all projected data sets coming from all source languages in our data set. The results are shown in Table 12. In all cases, we obtain the best score for cross-lingual dependency parsing so far which demonstrates the benefits of different projection algorithms. In our case, we only used a very simple concatenation approach and we expect that better combination techniques would work even better.

[2]Note that we have to leave out Swedish for this test as there is no test set available for this language.

	DE	EN	ES	FR	SV
LAS	**60.94**	**56.58**	**68.45**	**69.15**	**68.95**
UAS	67.89	63.89	75.33	74.75	76.48
LACC	79.02	73.26	81.99	83.18	80.85

Table 12: Combining projected data of all source languages to train target language parsing models. Additionally to LAS we also includes unlabeled attachment scores (UAS) and label accuracy (LACC).

5 Conclusions

In this paper, we propose several modifications and data sub-set selection techniques that can be used to improve the projection of syntactic annotation for cross-lingual dependency parsing. We show that it is beneficial to remove unnecessary dummy nodes from the projected trees and that it is useful to filter out sentences with uninformative annotation. These techniques lead to substantial improvements in labeled attachment scores when applied to automatically annotated bitexts and machine-translated text. We also introduce syntax-based SMT as yet another alternative to cross-lingual parsing and demonstrate its advantage over phrase-based models. Furthermore, a combination of projected resources leads to further gains and overall we present the highest scores for the cross-lingual parsing task so far.

There are several directions for future work. The most obvious question is related to data combination and multi-source transfer. A simple concatenation is certainly not optimal and more sophisticated data selection or instance weighting schemes are promising ideas for future research. Furthermore, the translation approach can be developed in various ways. First of all, we could look at improved translation that is optimized for the task of projection rather than translation quality. N-best lists could be explored as well and factored models may also help to improve the projection of POS tags.

References

Miguel Ballesteros and Joakim Nivre. 2012. MaltOptimizer: An Optimization Tool for MaltParser. In *Proceedings of EACL 2012*, pages 58–62.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Poster paper: Hunpos – an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 209–212, Prague, Czech Republic.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of ACL 2013*, pages 690–696.

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.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase Based Translation. In *Proceedings of NAACL-HLT 2003*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher J. Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007*, pages 177–180.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT Summit 2005*, pages 79–86.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-Source Transfer of Delexicalized Dependency Parsers. In *Proceedings of EMNLP 2011*, pages 62–72.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of ACL 2013*, pages 92–97.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective Sharing for Multilingual Dependency Parsing. In *Proceedings of ACL 2012*, pages 629–637.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of LREC 2006*, pages 2216–2219.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–52.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL 2003*, pages 160–167.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A Universal Part-of-Speech Tagset. In *Proceedings of LREC 2012*, pages 2089–2096.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual Word Clusters for Direct Transfer of Linguistic Structure. In *Proceedings of NAACL 2012*, pages 477–487.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target Language Adaptation of Discriminative Transfer Parsers. In *Proceedings of NAACL 2013*, pages 1061–1071.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. In *Proceedings of the 18th Conference Natural Language Processing and Computational Natural Language Learning (CoNLL)*, Baltimore, Maryland, USA.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of LREC 2012*, pages 2214–2218.

Jörg Tiedemann. 2014. Rediscovering annotation projection for cross-lingual parser induction. In *Proceedings of COLING 2014*, Dublin, Ireland, August.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing Multilingual Text Analysis Tools via Robust Projection Across Aligned Corpora. In *Proceedings of HLT 2011*, pages 1–8.