

# Low-Resource Syntactic Transfer with Unsupervised Source Reordering

**Mohammad Sadegh Rasooli**

Facebook AI  
Menlo Park, CA, USA  
rasooli@fb.com

**Michael Collins**

Department of Computer Science  
Columbia University  
mcollins@cs.columbia.edu

## Abstract

We describe a cross-lingual transfer method for dependency parsing that takes into account the problem of word order differences between source and target languages. Our model only relies on the Bible, a considerably smaller parallel data than the commonly used parallel data in transfer methods. We use the concatenation of projected trees from the Bible corpus, and the gold-standard treebanks in multiple source languages along with cross-lingual word representations. We demonstrate that reordering the source treebanks before training on them for a target language improves the accuracy of languages outside the European language family. Our experiments on 68 treebanks (38 languages) in the Universal Dependencies corpus achieve a high accuracy for all languages. Among them, our experiments on 16 treebanks of 12 non-European languages achieve an average UAS absolute improvement of 3.3% over a state-of-the-art method.

## 1 Introduction

There has recently been a great deal of interest in cross-lingual transfer of dependency parsers, for which a parser is trained for a target language of interest using treebanks in other languages. Cross-lingual transfer can eliminate the need for the expensive and time-consuming task of treebank annotation for low-resource languages. Approaches include annotation projection using parallel data sets (Hwa et al., 2005; Ganchev et al., 2009), direct model transfer through learning of a delexicalized model from other treebanks (Zeman and Resnik, 2008; Täckström et al., 2013), treebank translation (Tiedemann et al., 2014), using synthetic treebanks (Tiedemann and Agić, 2016; Wang and Eisner, 2016), using cross-lingual word representations (Täckström et al., 2012; Guo et al., 2016; Rasooli and Collins, 2017) and using cross-lingual dictionaries (Durrett et al., 2012).

Recent results from Rasooli and Collins (2017) have shown accuracies exceeding 80% on unlabeled attachment accuracy (UAS) for several European languages.<sup>1</sup> However non-European languages remain a significant challenge for cross-lingual transfer. One hypothesis, which we investigate in this paper, is that word-order differences between languages are a significant challenge for cross-lingual transfer methods. The main goal of our work is therefore to reorder gold-standard source treebanks to make those treebanks syntactically more similar to the target language of interest. We use two different approaches for source treebank reordering: 1) reordering based on dominant dependency directions according to the projected dependencies, 2) learning a classifier on the alignment data. We show that an ensemble of these methods with the baseline method leads to higher performance for the majority of datasets in our experiments. We show particularly significant improvements for non-European languages.<sup>2</sup>

The main contributions of this work are as follows:

- We propose two different syntactic reordering methods based on the dependencies projected using translation alignments. The first model is based on the dominant dependency direction in the target language according to the projected dependencies. The second model learns a reordering classifier from the small set of aligned sentences in the Bible parallel data.

<sup>1</sup>Specifically, Table 9 of Rasooli and Collins (2017) shows 13 datasets, and 11 languages, with UAS scores of over 80%; all of these datasets are in European languages.

<sup>2</sup>Specifically, performance of our method gives an improvement of at least 2.3% absolute scores in UAS on 11 datasets in 9 languages—Coptic, Basque, Chinese, Vietnamese, Turkish, Persian, Arabic, Indonesian Hebrew—with an average improvement of over 4.5% UAS.

- We run an extensive set of experiments on 68 treebanks for 38 languages. We show that by just using the Bible data, we are able to achieve significant improvements in non-European languages. Our ensemble method is able to maintain a high accuracy in European languages.
- We show that syntactic transfer methods can outperform a supervised model for cases in which the gold-standard treebank is very small. This indicates the strength of these models when the language is truly low-resource.

Unlike most previous work for which a simple delexicalized model with gold part-of-speech tags are used, we use lexical features and automatic part-of-speech tags. Our final model improves over two strong baselines, one with annotation projection and the other one inspired by the non-neural state-of-the-art model of [Rasooli and Collins \(2017\)](#). Our final results improve the performance on non-European languages by an average UAS absolute improvement of 3.3% and LAS absolute improvement of 2.4%.

## 2 Related Work

There has recently been a great deal of research on dependency parser transfer. Early work on direct model transfer ([Zeman and Resnik, 2008](#); [McDonald et al., 2011](#); [Cohen et al., 2011](#); [Rosa and Zabokrtsky, 2015](#); [Wang and Eisner, 2018a](#)) considered learning a delexicalized parser from one or many source treebanks. A number of papers ([Naseem et al., 2012](#); [Täckström et al., 2013](#); [Zhang and Barzilay, 2015](#); [Ammar et al., 2016](#); [Wang and Eisner, 2017](#)) have considered making use of topological features to overcome the problem of syntactic differences across languages. Our work instead reorders the source treebanks to make them similar to the target language before training on the source treebanks.

[Agić \(2017\)](#) use part-of-speech sequence similarity between the source and target language for selecting the source sentences in a direct transfer approach. [Ponti et al. \(2018\)](#) preprocess source trees to increase the isomorphy between the source and the target language dependency trees. They apply their method on a simple delexicalized model and their accuracy on the small set of languages that they have tried is significantly worse

than ours in all languages. The recent work by [Wang and Eisner \(2018b\)](#) reorders delexicalized treebanks of part-of-speech sequences in order to make it more similar to the target language of interest. The latter work is similar to our work in terms of using reordering. Our work is more sophisticated by using a full-fledged parsing model with automatic part-of-speech tags and every accessible dataset such as projected trees and multiple source treebanks as well as cross-lingual word embeddings for all languages.

Previous work ([Täckström et al., 2012](#); [Duong et al., 2015](#); [Guo et al., 2015, 2016](#); [Ammar et al., 2016](#)) has considered using cross-lingual word representations. A number of authors ([Durrett et al., 2012](#); [Rasooli and Collins, 2017](#)) have used cross-lingual dictionaries. We also make use of cross-lingual word representations and dictionaries in this paper. We use the automatically extracted dictionaries from the Bible to translate words in the source treebanks to the target language. One other line of research in the delexicalized transfer approach is creating a synthetic treebank ([Tiedemann and Agić, 2016](#); [Wang and Eisner, 2016, 2018b](#)).

Annotation projection ([Hwa et al., 2005](#); [Ganchev et al., 2009](#); [McDonald et al., 2011](#); [Ma and Xia, 2014](#); [Rasooli and Collins, 2015](#); [Lacroix et al., 2016](#); [Agić et al., 2016](#)) is another approach in parser transfer. In this approach, supervised dependencies are projected through word alignments and then used as training data. Similar to previous work ([Rasooli and Collins, 2017](#)), we make use of a combination of projected dependencies from annotation projection in addition to partially translated source treebanks. One other approach is treebank translation ([Tiedemann et al., 2014](#)) for which a statistical machine translation system is used to translate source treebanks to the target language. These models need a large amount of parallel data for having an accurate translation system.

Using the Bible data goes back to the work of [Diab and Finch \(2000\)](#) and [Yarowsky et al. \(2001\)](#). Recently there has been more interest in using the Bible data for different tasks, due to its availability for many languages ([Christodouloupoulos and Steedman, 2014](#); [Agić et al., 2015, 2016](#); [Rasooli and Collins, 2017](#)). Previous work ([Östling and Tiedemann, 2017](#)) has shown that the size of the Bible dataset does not provide a reliable machine

translation model. Previous work in the context of machine translation (Bisazza and Federico, 2016; Daiber et al., 2016) presumes the availability of a parallel data that is often much larger than the Bible data.

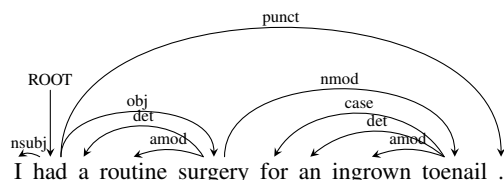
### 3 Baseline Model

Our model trains on the concatenation of projected dependencies  $\mathcal{P}$  and all of the source treebanks  $\mathcal{T}_1 \dots \mathcal{T}_k$ . The projected data is from the set of projected dependencies for which at least 80% of words have projected dependencies or there is a span of length  $l \geq 5$  such that all words in that span achieve a projected dependency. This is the same as the definition of dense structures  $\mathcal{P}_{80} \cup \mathcal{P}_{\geq 5}$  by Rasooli and Collins (2015).

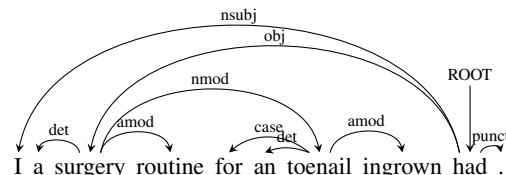
We use our reimplementation of the state-of-the-art neural biaffine graph-based parser of Dozat and Manning (2016)<sup>3</sup>. Because many words in the projected dependencies do not have a head assignment, the parser ignores words without heads during training. Inspired by Rasooli and Collins (2017), we replace every word in the source treebanks with its most frequent aligned translation word from the Bible data in the target language. If that word does not appear in the Bible, we use the original word. That way, we have a code-switched data for which some of the words are being translated. In addition to fine-tuning the word embeddings, we use the fixed pre-trained cross-lingual word embeddings using the training approach of Rasooli and Collins (2017) using the Wikipedia data and the Bible dictionaries.

### 4 Approach

Before making use of the source treebanks  $\mathcal{T}_1 \dots \mathcal{T}_k$  in the training data, we reorder each tree in the source treebanks to be syntactically more similar to the word order of the target language. In general, for a head  $h$  that has  $c$  modifiers  $m_1 \dots m_c$ , we decide to put each of the dependents  $m_i$  on the left or right of the head  $h$ . After placing them in the correct side of the head, the order in the original source sentence is preserved. Figure 1 shows a real example of an English tree that is reordered for the sake of Persian as the target language. Here we see that we have a verb-final sentence, with nominal modifiers following



(a) Original tree.



(b) Persian-specific reordered tree.

Figure 1: An example of a gold-standard English tree that is reordered to look similar to the Persian syntactic order.

the head noun. If one aims to translate this English sentence word by word, the reordered sentence gives a very good translation without any change in the sentence.

As mentioned earlier, we use two different approaches for source treebank reordering: 1) reordering based on dominant dependency directions according to the projected dependencies, 2) learning a classifier on the alignment data. We next describe these two methods.

#### 4.1 Model 1: Reordering Based on Dominant Dependency Direction

The main goal of this model is to reorder source dependencies based on dominant dependency directions in the target language. We extract dominant dependency directions according to the projected dependencies  $\mathcal{P}$  from the alignment data, and use the information for reordering source treebanks.

Let the tuple  $\langle i, m, h, r \rangle$  show the dependency of the  $m$ 'th word in the  $i$ 'th projected sentence for which the  $h$ 'th word is the parent with the dependency label  $r$ .  $\langle i, m, \text{NULL}, \text{NULL} \rangle$  shows an unknown dependency for the  $m$ 'th word: this occurs when some of the words in the target sentence do not achieve a projected dependency. We use the notations  $h(i, m)$  and  $r(i, m)$  to show the head index and dependency label of the  $m$ 'th word in the  $i$ 'th sentence.

**Definition 1** *Dependency direction:*  $d(i, m)$  shows the dependency direction of the  $m$ 'th

<sup>3</sup><https://github.com/rasoolims/universal-parser>

modifier word in the  $i$ 'th sentence:

$$d(i, m) = \begin{cases} 1 & \text{if } h(i, m) > m \\ -1 & \text{otherwise} \end{cases}$$

**Definition 2** *Dependency direction proportion:* Dependency direction proportion of each dependency label  $l$  with direction  $d \in \{-1, 1\}$  is defined as:

$$\alpha^{(\mathcal{P})}(l, d) = \frac{\sum_{i=1}^{|\mathcal{P}|} \sum_{m=1}^{|\mathcal{P}^{(i)}|} \mathbb{I}(r(i, m) = l \ \& \ d(i, m) = d)}{\sum_{i=1}^{|\mathcal{P}|} \sum_{m=1}^{|\mathcal{P}^{(i)}|} \mathbb{I}(r(i, m) = l)}$$

**Definition 3** *Dominant dependency direction:* For each dependency label  $l$ , we define the dominant dependency direction  $\lambda^{(\mathcal{P})}(l) = d$  if  $\alpha^{(\mathcal{P})}(l, d) > 0.75$ . In cases where there is no dominant dependency direction,  $\lambda^{(\mathcal{P})}(l) = 0$ .

We consider the following dependency labels for extracting dominant dependency direction information: nsubj, obj, iobj, csubj, ccomp, xcomp, obl, vocative, expl, dislocated, advcl, advmod, aux, cop, nmod, appos, nummod, acl, amod. We find the direction of other dependency relations, such as most of the function word dependencies and other non-core dependencies such as conjunction, not following a fixed pattern in the Universal Dependencies corpus.

**Reordering condition** Given a set of projections  $\mathcal{P}$ , we calculate the dominant dependency direction information for the projections  $\lambda^{(\mathcal{P})}$ . Similar to the projected dependencies, we extract *supervised* dominant dependency directions from the gold-standard source treebank  $\mathcal{D}$ :  $\lambda^{(\mathcal{D})}$ . When we encounter a gold-standard dependency relation  $\langle i, m, h, r \rangle$  in a source treebank  $\mathcal{D}$ , we change the direction if the following condition holds:

$$\lambda^{(\mathcal{D})}(r) \neq \lambda^{(\mathcal{P})}(r) \ \text{and} \ \lambda^{(\mathcal{P})}(r) = -d(i, m)$$

In other words, if the source and target languages do not have the same dominant dependency direction for  $r$  and the dominant direction of the target language is the reverse of the current direction, we change the direction of that dependency. Reordering multiple dependencies in a gold standard tree then results in a reordering of the full tree, as for example in the transformation from Figure 1a to Figure 1b.

## 4.2 Model 2: Reordering Classifier

We now describe our approach for learning a reordering classifier for a target language using the alignment data. Unlike the first model for which we learn concrete rules, this model learns a reordering classifier from automatically aligned data. This model has two steps; the first step prepares the training data from the automatically aligned parallel data, and the second step learns a classifier from the training data.

### 4.2.1 Preparing Training Data from Alignments

The goal of this step is to create training data for the reordering classifier. This data is extracted from the concatenation of parallel data from all source languages translated to the target language. Given a parallel dataset  $(e^{(i)}, f^{(i)})$  for  $i = 1 \dots n$  that contains pairs of source and target sentences  $e^{(i)}$  and  $f^{(i)}$ , the following steps are applied to create training data:

1. **Extracting reordering mappings from alignments:** We first extract intersected word alignments for each source-target sentence pair. This is done by running the Giza++ alignments (Och and Ney, 2003) in both directions. We ignore sentence pairs that more than half of the source words do not get alignment. We create a new mapping  $\mu^{(i)} = \mu_1^{(i)} \dots \mu_{s_i}^{(i)}$  that maps each index  $1 \leq j \leq s_i$  in the original source sentence to a unique index  $1 \leq \mu_j^{(i)} \leq s_i$  in the reordered sentence.
2. **Parsing source sentences:** We parse each source sentence using the supervised parser of the source language. We use the mapping  $\mu^{(i)}$  to come up with a reordered tree for each sentence. In cases for which the number of non-projective arcs in the projected tree increase compared to the original tree, we do not use the sentence in the final training data.
3. **Extracting classifier instances:** We create a training instance for every modifier word  $\langle i, m, h, r \rangle$ . The decision about the direction of each dependency can be made based on the following condition:

$$d^*(i, m) = \begin{cases} 1 & \text{if } \mu_h^{(i)} > \mu_m^{(i)} \\ -1 & \text{otherwise} \end{cases}$$



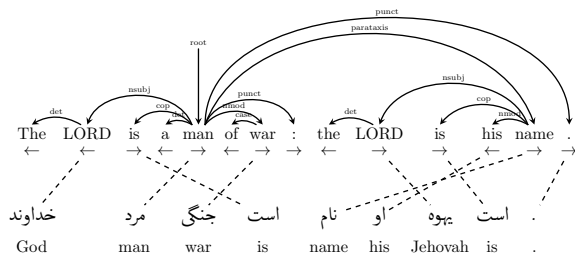


Figure 2: A reordering example from the Bible for English-Persian language pair. The Persian words are written from left to right for the ease of presentation. The arrows below the English words show the new dependency direction with respect to the word alignments to the Persian side. The reordered sentence would be “The LORD a man of war is : his name the LORD is .”.

In other words, we decide about the new order of a dependency according to the mapping  $\mu^{(i)}$ .

Figure 2 shows an example for the data preparation step. As shown in the figure, the new directions for the English words are decided according to the Persian alignments.

#### 4.2.2 Classifier

The reordering classifier decides about the new direction of each dependency according to the recurrent representation of the head and dependent words. For a source sentence  $e^{(i)} = e_1^{(i)} \dots e_{s_i}^{(i)}$  that belongs to a source language  $\mathcal{L}$ , we first obtain its recurrent representation  $\eta^{(i)} = \eta_1^{(i)} \dots \eta_{s_i}^{(i)}$  by running a deep (3 layers) bi-directional LSTM (Hochreiter and Schmidhuber, 1997), where  $\eta_j^{(i)} \in \mathbb{R}^{d_h}$ . For every dependency tuple  $\langle i, m, h, r \rangle$ , we use a multi-layer Perceptron (MLP) to decide about the new order  $dir \in \{-1, 1\}$  of the  $m$ 'th word with respect to its head  $h$ :

$$p(dir|i, m, h, r) = \text{softmax}_{dir}(W\phi(i, m, h, r))$$

where  $W \in \mathbb{R}^{2 \times d_\phi}$  and  $\phi(i, m, h, r) \in \mathbb{R}^{d_\phi}$  is as follows:

$$\phi(i, m, h, r) = \text{relu}(Hq(i, m, h, r) + B)$$

where  $\text{relu}$  is the rectified linear unit activation (Nair and Hinton, 2010),  $H \in \mathbb{R}^{d_\phi \times d_q}$ ,  $B \in \mathbb{R}^{d_\phi}$ , and  $q(i, m, h, r) \in \mathbb{R}^{d_q}$  is as follows:

$$q(i, m, h, r) = [\eta_m^{(i)}; \eta_h^{(i)}; R[r]; \Lambda[\mathbb{I}(h > m)]; L[\mathcal{L}]$$

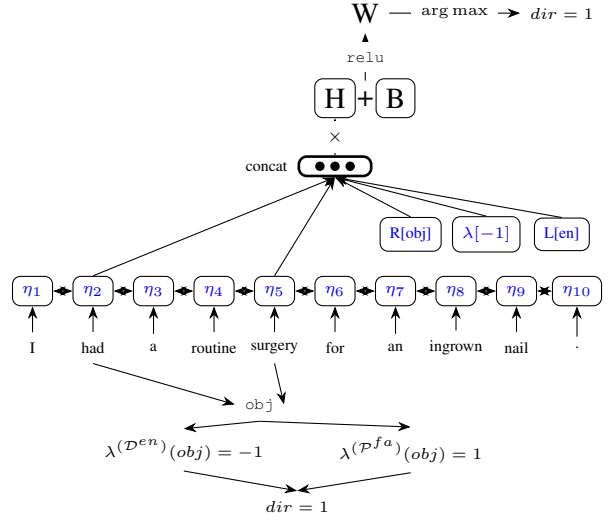


Figure 3: Two different approaches for reordering the dependency order for the example in Figure 1. The reordering classifier is shown on top, for the dependency relation between the words “had” and “surgery” with an “obj” relation. At the bottom, the reordering model based on dominant dependency direction information is shown.

where  $\eta_m^{(i)}$  and  $\eta_h^{(i)}$  are the recurrent representations for the modifier and head words respectively,  $R$  is the dependency relation embedding dictionary that embeds every dependency relation to a  $\mathbb{R}^{d_r}$  vector,  $\Lambda$  is the direction embedding for the original position of the head with respect to its head and embeds each direction to a 2-dimensional vector, and  $L$  is the language embedding dictionary that embeds the source language  $\mathcal{L}$  to a  $\mathbb{R}^{d_L}$  vector.

The input to the recurrent layer is the concatenation of two input vectors. The first vector is the sum of the fixed pre-trained cross-lingual embeddings, and randomly initialized word vector. The second vector is the part-of-speech tag embeddings.

Figure 3 shows a graphical depiction of the two reordering models that we use in this work.

## 5 Experiments

**Datasets and Tools** We use 68 datasets from 38 languages in the Universal Dependencies corpus version 2.0 (Nivre et al., 2017). The languages are Arabic (ar), Bulgarian (bg), Coptic (cop), Czech (cs), Danish (da), German (de), Greek (el), English (en), Spanish (es), Estonian (et), Basque (eu), Persian (fa), Finnish (fi), French (fr), Hebrew (he), Hindi (hi), Croatian (hr), Hungarian

(hu), Indonesian (id), Italian (it), Japanese (ja), Korean (ko), Latin (la), Lithuanian (lt), Latvian (lv), Dutch (nl), Norwegian (no), Polish (pl), Portuguese (pt), Romanian (ro), Russian (ru), Slovak (sk), Slovene (sl), Swedish (sv), Turkish (tr), Ukrainian (uk), Vietnamese (vi), and Chinese (zh).

We use the Bible data from [Christodouloupoulos and Steedman \(2014\)](#) for the 38 languages. We extract word alignments using Giza++ default model ([Och and Ney, 2003](#)). Following [Rasooli and Collins \(2015\)](#), we obtain intersected alignments and apply soft POS consistency to filter potentially incorrect alignments. We use the Wikipedia dump data to extract monolingual data for the languages in order to train monolingual embeddings. We follow the method of [Rasooli and Collins \(2017\)](#) to use the extracted dictionaries from the Bible and monolingual text from Wikipedia to create cross-lingual word embeddings. We use the UDPipe pretrained models ([Straka and Straková, 2017](#)) to tokenize Wikipedia, and a reimplement of the Perceptron tagger of [Collins \(2002\)](#)<sup>4</sup> to achieve automatic POS tags trained on the training data of the Universal Dependencies corpus ([Nivre et al., 2017](#)). We use word2vec ([Mikolov et al., 2013](#))<sup>5</sup> to achieve embedding vectors both in monolingual and cross-lingual settings.

**Supervised Parsing Models** We trained our supervised models on the union of all datasets in a language to obtain a supervised model for each language. It is worth noting that there are two major changes that we make to the neural parser of [Dozat and Manning \(2016\)](#) in our implementation<sup>6</sup> using the Dynet library ([Neubig et al., 2017](#)): first, we add a one-layer character BiLSTM to represent the character information for each word. The final character representation is obtained by concatenating the forward representation of the last character and the backward representation of the first character. The concatenated vector is summed with the randomly initialized as well as fixed pre-trained cross-lingual word embedding vectors. Second, inspired by [Weiss et al. \(2015\)](#), we maintain the moving average parameters to obtain more robust parameters at decoding time.

<sup>4</sup><https://github.com/rasoolims/SemiSupervisedPosTagger>

<sup>5</sup><https://github.com/dav/word2vec>

<sup>6</sup><https://github.com/rasoolims/universal-parser>

We excluded the following languages from the set of source languages for *annotation projection* due to their low supervised accuracy: Estonian, Hungarian, Korean, Latin, Lithuanian, Latvian, Turkish, Ukrainian, Vietnamese, and Chinese.

**Baseline Transfer Models** We use two baseline models: 1) Annotation projection: This model only trains on the projected dependencies. 2) Annotation projection + direct transfer: To speed up training, we sample at most thousand sentences from each treebank, comprising a training data of about 37K sentences.

### 5.1 Reordering Ensemble Model

We noticed that our reordering models perform better in non-European languages, and perform slightly worse in European languages. We use the following ensemble model to make use of all of the three models (annotation projection + direct transfer, and the two reordering models), to make sure that we always obtain an accurate parser.

The ensemble model is as follows: given three output trees for the  $i$ 'th sentence  $\langle i_j, m, h_j, r_j \rangle$  for  $j = 1, 2, 3$  in the target language  $\mathcal{L}$ , where the first tuple ( $j = 1$ ) belongs to the baseline model, the second ( $j = 2$ ) and third ( $j = 3$ ) belong to the two reordering models, we weight each dependency edge with respect to the following conditions:

$$\omega(m, h, r) = z(m, h, r) \cdot \sum_{j=1}^3 c(j, \mathcal{L}) \cdot \mathbb{I}(\langle i_j, m, h, r \rangle)$$

where  $c(j, \mathcal{L})$  is a coefficient that puts more weight on the first or the other two outputs depending on the target language family:

$$c(j, \mathcal{L}) = \begin{cases} 2 & \text{if } j = 1 \text{ \& } \mathcal{L} \text{ is European} \\ 2 & \text{if } j > 1 \text{ \& } \\ & \mathcal{L} \text{ is not European} \\ 1 & \text{otherwise} \end{cases}$$

and  $z(m, h, r)$  is a simple weighting depending on the dominant order information:

$$z(m, h, r) = \begin{cases} 1 & \text{if } \text{dir}(\langle m, h \rangle) = -\lambda^{(\mathcal{P})}(r) \\ 3 & \text{if } \text{dir}(\langle m, h \rangle) = \lambda^{(\mathcal{P})}(r) \\ 2 & \text{otherwise } (\lambda^{(\mathcal{P})}(r) = 0) \end{cases}$$

Variable	Notation	Size
Word embedding	$d_w$	100
POS embedding	$d_p$	100
Bi-LSTM	$d_h$	400
Dep. relation embedding	$d_r$	50
Language ID embedding	$d_L$	50
Hidden layer	$d_\phi$	200
Number of BiLSTM layers	–	3
Mini-batch size (tokens)	–	$\sim 1000$

Table 1: Parameter values in the reordering classifier model.

The above coefficients are modestly tuned on the Persian language as our development language. We have not seen any significant change in modifying the numbers: instead, the fact that an arc with a dominant dependency direction is regarded as a more valuable arc, and the baseline should have more effect in the European languages suffices for the ensemble model.

We run the Eisner first-order graph-based algorithm (Eisner, 1996) on top of the edge weights  $\omega$  to extract the best possible tree.

## 5.2 Parameters

We run all of the transfer models with 4000 mini-batches, in which each mini-batch contains approximately 5000 tokens. We follow the same parameters as in Dozat and Manning (2016) and use a dimension of 100 for character embeddings. For the reordering classifier, we use the Adam algorithm (Kingma and Ba, 2014) with default parameters to optimize the log-likelihood objective. We filter the alignment data to keep only those sentences for which at least half of the source words have an alignment. We randomly choose 1% of the reordering data as our heldout data for deciding when to stop training the reordering models. Table 1 shows the parameter values that we use in the reordering classifier.

## 5.3 Results

Table 2 shows the results on the Universal Dependencies corpus (Nivre et al., 2017). As shown in the table, the algorithm based on dominant dependency directions improves the accuracy on most of the non-European languages and performs slightly worse than the baseline model in the European languages. The ensemble model, in spite of its simplicity, improves over the baseline in most of

the languages, leading to an average UAS improvement of 0.9 for all languages and 3.3 for non-European languages. This improvement is very significant in many of the non-European languages; for example, from an LAS of 37.6 to 52.7 in Coptic, from a UAS of 44.9 to 53.7 in Basque, from a UAS of 40.6 to 47.0 in Chinese. Our model also outperforms the supervised models in Ukrainian and Latvian. That is an interesting indicator that for cases that the training data is very small for a language (37 sentences for Ukrainian, and 153 sentences for Latvian), our transfer approach outperforms the supervised model.

## 6 Analysis

In this section, we briefly describe our analysis based on the results in the ensemble model and the baseline. For some languages such as Coptic, the number of dense projected dependencies is too small (two trees) such that the parser gives a worse learned model than a random baseline. For some other languages, such as Norwegian and Spanish, this number is too high (more than twenty thousand trees), such that the baseline model performs very well.

The dominant dependency direction model generally performs better than the classifier. Our manual investigation shows that the classifier kept many of the dependency directions unchanged, while the dominant dependency direction model changed more directions. Therefore, the dominant direction model gives a higher recall with the expense of losing some precision. The training data for the reordering classifier is very noisy due to wrong alignments. We believe that the dominant direction model, besides its simplicity, is a more robust classifier for reordering, though the classifier is helpful in an ensemble setting.

Our detailed analysis show that we are able to improve the head dependency relation for the three most important head POS tags in the dependency grammar. We see that this improvement is more consistent for all non-European languages. Table 3 shows the differences in parsing f-score of dependency relations for adjectives, nouns and verbs as the head. As we see in the Table, we are able to improve the head dependency relation for the three most important head POS tags in the dependency grammar. We see that this improvement is more consistent for all non-European languages. We skip the details of those analysis due to space

Dataset	Baselines				Reordering								Supervised	
	Projection		Direct+Proj		Dominant		Classifier		Ensemble		Difference			
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Coptic	2.0	0.4	58.5	37.6	69.1	52.7	65.5	50.9	69.6	52.7	11.1	15.1	86.9	80.1
Basque	39.5	22.0	44.9	29.0	53.7	34.0	48.6	32.2	53.7	34.4	8.8	5.4	81.9	75.9
Chinese	23.6	10.8	40.6	17.8	47.3	25.4	45.4	23.5	47.0	25.6	6.4	7.8	81.1	74.8
Vietnamese	44.6	26.8	51.2	33.6	55.3	34.5	50.4	34.2	55.1	34.5	4.0	0.9	66.2	56.7
Turkish_pud	44.7	19.9	46.6	24.5	50.3	26.7	42.6	22.0	49.9	26.3	3.4	1.8	56.7	31.7
Persian	54.4	46.2	61.8	53.0	64.3	54.7	63.0	53.4	65.1	55.4	3.3	2.4	87.8	83.6
Arabic_pud	60.3	44.2	65.2	50.5	68.2	52.0	66.5	51.4	68.3	52.3	3.2	1.8	71.9	58.8
Indonesian	59.9	42.8	72.1	56.0	73.6	56.5	72.9	56.8	74.6	56.7	2.5	0.6	84.8	77.4
Turkish	44.6	23.9	46.6	29.3	48.9	30.6	44.9	26.6	49.0	30.0	2.4	0.7	64.2	52.5
Hebrew	63.1	46.9	70.4	55.4	72.4	54.9	71.6	55.7	72.7	55.4	2.3	0.0	88.2	82.4
Arabic	49.5	36.8	58.9	46.8	60.8	48.3	59.2	46.9	61.2	48.8	2.3	2.0	85.6	78.9
Japanese	54.8	38.9	65.2	46.5	65.9	46.8	64.1	44.8	66.6	46.8	1.4	0.3	94.5	92.7
Japanese_pud	58.6	44.1	66.8	51.5	67.4	51.5	64.7	48.4	67.9	51.9	1.1	0.4	94.7	93.5
Korean	34.3	17.3	43.0	24.8	43.5	23.8	43.6	26.4	44.1	24.7	1.1	-0.2	76.2	69.9
Hindi_pud	53.4	43.3	58.2	47.6	58.3	47.5	58.8	48.5	58.9	48.2	0.6	0.6	70.2	55.6
Lithuanian	60.6	42.5	66.6	49.5	63.7	46.8	64.6	46.0	67.2	49.9	0.6	0.4	54.8	40.0
Czech_cac	33.9	14.8	76.2	66.9	76.3	66.7	75.2	65.8	76.7	67.4	0.5	0.6	92.1	88.3
Czech_cltt	13.7	5.1	69.4	59.7	69.7	59.5	66.6	57.8	70.0	60.3	0.5	0.6	88.9	84.9
French_partut	81.6	75.2	84.3	77.8	84.9	78.4	84.4	78.1	84.8	78.4	0.5	0.5	90.0	85.1
Croatian	70.6	59.9	79.4	69.9	79.3	69.5	77.9	67.7	79.9	70.1	0.5	0.2	86.8	80.4
Greek	62.3	47.2	75.9	63.9	75.4	63.1	74.7	62.5	76.4	64.1	0.4	0.2	88.0	84.4
Russian_pud	75.7	65.8	81.1	72.2	80.9	72.2	79.9	70.7	81.5	72.7	0.4	0.5	86.5	74.1
German	71.4	62.3	75.4	67.1	75.6	67.1	75.5	66.4	75.8	67.3	0.4	0.2	85.9	81.2
French	80.2	72.9	83.0	75.9	82.9	75.9	83.3	75.9	83.4	76.2	0.4	0.3	90.4	86.9
Czech	33.9	14.5	74.6	65.3	74.1	64.4	73.0	63.7	75.0	65.8	0.4	0.5	92.5	89.1
Finnish_pud	64.1	52.5	67.2	55.0	66.8	55.0	67.3	55.1	67.5	55.5	0.4	0.5	81.6	74.5
Dutch	59.2	48.2	68.5	55.2	69.6	55.9	68.3	54.4	68.8	55.4	0.4	0.1	83.5	76.6
Russian	68.9	59.4	75.1	63.9	75.4	64.1	74.5	63.4	75.5	64.3	0.4	0.4	85.7	77.9
Latin_littb	56.4	42.5	63.0	49.2	63.2	49.5	62.4	48.7	63.3	49.7	0.4	0.4	89.5	86.5
Norwegian_nynorsk	72.5	62.9	76.4	68.1	76.5	68.0	76.1	67.3	76.8	68.4	0.3	0.3	91.3	88.8
Ukrainian	55.1	36.9	64.3	46.1	64.5	45.7	61.7	42.2	64.6	45.9	0.3	-0.2	43.3	22.1
Bulgarian	80.4	69.4	83.8	73.8	84.0	73.8	83.1	73.0	84.1	73.9	0.3	0.1	90.9	86.0
English_lines	75.6	66.5	77.8	69.0	78.9	69.9	77.0	68.2	78.1	69.2	0.3	0.3	85.8	80.5
Finnish_ftb	63.9	46.5	66.0	48.3	65.8	47.6	65.7	48.1	66.3	48.4	0.3	0.1	81.1	74.4
Russian_syntagrus	69.4	57.5	73.9	62.2	73.8	61.8	73.2	61.2	74.2	62.3	0.3	0.1	91.3	88.3
Finnish	60.6	48.7	64.6	51.9	63.5	51.2	63.7	51.1	64.8	52.0	0.2	0.1	80.9	73.5
Hungarian	58.3	41.1	67.8	49.0	67.8	48.9	65.8	47.4	68.0	49.1	0.2	0.1	78.2	69.8
Czech_pud	35.7	16.6	77.5	69.3	76.7	67.6	76.2	67.7	77.7	69.4	0.2	0.2	89.9	84.4
Dutch_lassysmall	61.8	52.1	73.9	63.4	73.8	62.8	73.0	61.9	74.0	63.3	0.2	0.0	91.3	87.3
Slovenian_sst	58.4	44.1	61.7	47.7	61.6	47.7	61.6	47.4	61.9	48.0	0.2	0.3	70.6	63.6
English_pud	73.5	65.5	75.9	69.3	77.1	69.9	74.5	67.7	76.0	69.4	0.2	0.2	88.3	84.2
German_pud	74.1	65.3	77.8	68.9	77.7	68.5	76.9	67.4	78.0	68.8	0.1	0.0	85.9	79.0
Polish	77.6	64.7	79.9	67.9	79.7	67.5	79.5	67.2	80.1	68.0	0.1	0.1	89.4	83.3
Swedish_lines	77.2	67.7	81.1	71.6	80.7	71.1	80.1	70.4	81.3	71.7	0.1	0.1	86.9	81.5
English	70.1	61.6	72.8	64.6	73.5	65.2	71.6	63.5	72.9	64.8	0.1	0.3	88.2	84.8
Spanish	78.5	68.0	83.1	73.8	83.2	73.8	82.3	72.8	83.2	73.9	0.1	0.1	89.3	83.9
Swedish	75.3	67.0	79.0	70.9	78.8	70.9	78.2	70.0	79.1	71.0	0.1	0.1	86.7	82.3
English_partut	72.0	65.3	77.4	71.1	78.0	71.1	76.3	69.9	77.5	71.2	0.1	0.1	88.4	83.0
Swedish_pud	75.9	67.4	80.5	72.1	80.2	72.0	79.2	71.0	80.6	72.1	0.1	0.0	84.0	77.6
Italian	81.3	74.4	85.0	79.0	85.4	79.5	84.4	78.1	85.1	79.1	0.1	0.0	92.1	89.5
Romanian	72.8	59.0	76.8	64.2	76.2	63.7	75.3	63.2	76.8	64.3	0.1	0.1	89.6	83.5
Estonian	63.1	40.8	66.7	46.0	65.6	45.8	65.5	45.2	66.7	46.1	0.1	0.2	71.6	60.7
Portuguese	62.6	50.7	84.1	76.9	83.7	76.6	83.4	76.2	84.2	77.1	0.0	0.2	90.6	85.6
Portuguese_br	60.6	47.7	81.3	71.2	80.8	70.8	80.8	70.4	81.4	71.3	0.0	0.2	91.6	89.0
Norwegian_bokmaal	78.0	70.5	80.5	73.2	80.6	73.4	79.7	72.1	80.5	73.2	0.0	0.0	92.1	89.7
French_pud	81.0	72.8	83.7	75.7	84.2	76.2	83.3	75.2	83.7	75.7	0.0	0.0	89.1	83.8
Spanish_pud	81.3	70.9	84.3	75.6	84.6	76.0	83.6	74.6	84.3	75.7	0.0	0.1	89.1	80.8
Latvian	59.0	43.6	63.3	47.2	62.1	45.6	60.7	44.7	63.3	47.0	0.0	-0.2	71.3	61.2
Italian_pud	83.8	76.0	87.3	81.3	87.5	81.3	86.5	79.9	87.3	81.2	0.0	-0.1	91.9	88.4
French_sequoia	79.1	73.0	82.2	76.4	81.6	75.8	81.9	76.0	82.2	76.4	0.0	0.0	90.4	86.7
Latin	49.2	33.6	53.9	36.2	51.3	33.3	54.0	35.5	53.9	35.4	0.0	-0.8	67.2	54.5
Slovene	76.4	67.6	82.1	74.2	81.3	73.0	81.3	73.3	82.0	74.2	-0.1	0.0	88.9	85.4
Spanish_ancora	77.7	66.2	82.4	72.7	82.0	72.2	81.4	71.3	82.3	72.5	-0.1	-0.3	91.1	87.0
Danish	70.7	61.7	75.7	67.4	75.3	66.7	74.6	66.2	75.6	67.2	-0.1	-0.2	83.1	79.3
Portuguese_pud	63.5	51.8	82.7	75.8	82.5	75.8	82.0	74.8	82.6	75.7	-0.2	-0.1	86.4	78.5
Latin_proiel	59.2	46.2	61.5	47.4	60.9	47.1	60.2	46.0	61.3	47.2	-0.2	-0.2	80.9	75.4
Slovak	73.6	63.8	78.7	71.0	78.0	69.8	77.1	68.7	78.5	70.7	-0.2	-0.3	83.5	77.9
Hindi	58.7	47.2	63.7	50.0	62.3	49.0	62.6	49.3	62.7	49.4	-1.0	-0.6	94.2	90.4
Avg. All	62.0	49.7	71.2	59.3	71.7	59.6	70.6	58.7	<b>72.1</b>	<b>60.0</b>	0.9	0.7	83.9	77.3
Avg. Non-EU	46.6	32.0	57.1	40.9	60.1	43.1	57.8	41.9	<b>60.4</b>	<b>43.3</b>	3.3	2.4	80.3	72.2

Table 2: Dependency parsing results, in terms of unlabeled attachment accuracy (UAS) and labeled attachment accuracy (LAS) after ignoring punctuations, on the Universal Dependencies v2 test sets (Nivre et al., 2017) using supervised part-of-speech tags. The results are sorted by their “difference” between the ensemble model and the baseline. The rows for non-European languages are highlighted with cyan. The rows that are highlighted by pink are the ones that the transfer model outperforms the supervised model. For all of the non-European datasets except “hi”, our model outperforms significantly better in terms of UAS with  $p < 0.001$  using McNemar’s test.



data	ADJ		NOUN		VERB	
	Base	Ens.	Base	Ens.	Base	Ens.
ar	40.4	46.7	70.6	72.5	55.3	58.8
ar_pud	32.3	39.7	73.2	75.9	67.2	70.1
bg	70.6	71.1	85.8	86.2	86.2	86.5
cop	0.0	0.0	63.4	75.7	64.6	76.4
cs	64.8	64.9	77.9	78.5	76.5	76.7
cs_cac	66.0	65.7	79.7	80.5	77.3	77.6
cs_cltt	55.9	56.5	76.9	77.7	68.3	68.9
cs_pud	71.2	70.9	79.4	80	80.2	80.3
da	70.9	71.2	79.5	79.3	79.5	79.6
de	65.7	66.7	81.3	81.5	75.8	76.3
de_pud	61.3	62.4	81.5	81.5	81.0	81.2
el	64.3	64.8	79.8	80.5	75.6	75.8
en	77.7	78.8	70.6	70.4	81.0	81.3
en_lines	74.4	74.7	78.3	78.5	82.2	82.8
en_partut	71.9	72.1	76.6	76.7	82.6	82.7
en_pud	69.5	70.6	75.4	75.5	81.2	81.6
es	75.6	74.6	88.0	88.4	80.6	80.9
es_ancora	71.3	71.4	87.4	87.4	83.0	82.9
es_pud	66.5	66.3	89.0	89.1	83.2	83.2
et	59.5	59.6	59.6	59.5	75.4	75.5
eu	31.1	35.4	37.6	47.9	52.4	61.2
fa	46.2	51.6	68.7	70.7	53.7	59.7
fi	65.8	66.3	61.8	62.5	70.5	70.5
fi_ftb	64.7	65.5	64.7	65.1	69.2	69.5
fi_pud	58.1	59.4	63.8	64.1	74.6	74.8
fr	74.1	74.6	87.3	87.5	81.9	82.7
fr_partut	72.2	72.9	88.4	88.8	83.1	83.8
fr_pud	71.3	71.1	88.7	88.8	81.0	81.1
fr_sequoia	72.0	72.0	86.5	86.6	82.2	82.0
he	64.7	69.1	75.6	77.8	68.1	70.6
hi	22.3	23.5	75.9	74.9	57.5	57.9
hi_pud	48.1	49.3	67.8	67.9	56.6	58.7
hr	72.3	71.8	82.2	82.4	83.1	83.8
hu	42.5	43.3	71.8	72.5	73.6	73.7
id	63.2	67.3	70.7	74.5	78.0	79.7
it	61.4	63.3	89.1	89.1	85.2	85.4
it_pud	71.7	72.0	90.7	90.7	87.1	87.2
ja	52.8	59.5	73.1	74.6	65.1	66.5
ja_pud	60.4	65.4	71.5	72.6	66.7	68.3
ko	55.7	52.9	23.5	24.3	52.4	54.3
la	35.1	35.6	43.8	44.4	58.8	58.5
la_itb	57.9	57.4	65.5	66.5	63.5	63.6
la_proiel	55.2	55.4	61.8	61.6	64.3	64.1
lt	54.0	57.1	70.8	72.2	69.7	69.7
lv	58.7	60.2	57.0	57.2	70.3	70.6
nl	57.7	61.3	81.9	81.7	66.4	67.2
nl_lassy_small	46.4	47.6	79.8	80.0	75.4	75.3
no_bokmaal	76.0	75.9	83.4	83.4	84.2	84.4
no_nynorsk	69.7	70.5	81.4	81.8	79.6	79.9
pl	66.1	67.2	79.2	79.4	85.2	85.2
pt	72.1	73.3	88.7	88.8	82.9	82.7
pt_br	39.5	39.5	88.2	88.2	77.8	77.7
pt_pud	61.4	60.5	89.0	88.8	81.2	81.2
ro	55.6	56.4	79.3	79.5	80.3	80.3
ru	52.3	53.1	77.9	78.5	79.8	80.0
ru_pud	64.4	64.5	83.1	83.7	81.9	82.4
ru_syntagrus	57.3	56.9	78.7	79.2	74.3	74.6
sk	69.5	69.5	80.5	80.3	84.0	83.5
sl	73.6	72.6	83.3	83.4	85.2	85.2
sl_sst	60.6	61.6	69.4	69.6	67.4	67.1
sv	77.0	76.9	82.8	82.9	81.1	81.4
sv_lines	78.7	78.9	85.1	85.1	83.1	83.3
sv_pud	77.2	77.3	83.4	83.4	83.8	84.1
tr	42.3	46.8	49.4	47.9	48.0	51.5
tr_pud	43.2	46.7	50.0	52.0	49.5	53.5
uk	49.3	48.8	64.1	64.6	71.9	72.3
vi	31.5	35.7	50.6	56.5	55.1	58.3
zh	47.7	52.1	47.5	56.4	43.0	45.7

Table 3: Unlabeled attachment f-score of POS tags as heads for the baseline and the reordering ensemble model. As shown in the table, our model improves over the baseline in most cases.

limitations. More thorough analysis can be found in (Rasooli, 2019, Chapter 6).

For a few number of languages such as Vietnamese, the best model, even though improves

over a strong baseline, still lacks enough accuracy to be considered as a reliable parser in place of a supervised model. We believe that more research on those language will address the mentioned problem. Our current model relies on supervised part-of-speech tags. Future work should study using transferred part-of-speech tags instead of supervised tags, leading to a much more realistic scenario for low-resource languages.

We have also calculated the POS trigram cosine similarity between the target language gold standard treebanks, and the three source training datasets (original, and the two reordered datasets). In all of the non-European languages, the cosine similarity of the reordered datasets improved with different values in the range of (0.002, 0.02). For Czech, Portuguese, German, Greek, English, Romanian, Russian, and Slovak, both of the reordered datasets slightly decreased the trigram cosine similarity. For other languages, the cosine similarity was roughly the same.

## 7 Conclusion

We have described a cross-lingual dependency transfer method that takes into account the problem of word order differences between the source and target languages. We have shown that applying projection-driven reordering improves the accuracy of non-European languages while maintaining the high accuracies in European languages. The focus of this paper is primarily of dependency parsing. Future work should investigate the effect of our proposed reordering methods on truly low-resource machine translation.

## Acknowledgements

We deeply thank the anonymous reviewers for their useful feedback and comments.

## References

- Željko Agić. 2017. [Cross-lingual parser selection for low-resource languages](#). In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 1–10. Association for Computational Linguistics.
- Željko Agić, Dirk Hovy, and Anders Søgaard. 2015. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Pro-*

- cessing (*Volume 2: Short Papers*), volume 2, pages 268–272.
- Željko Agić, Anders Johannsen, Barbara Plank, Héctor Alonso Martínez, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.
- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. [Many languages, one parser](#). *Transactions of the Association for Computational Linguistics*, 4:431–444.
- Arianna Bisazza and Marcello Federico. 2016. A survey of word reordering in statistical machine translation: Computational models and language phenomena. *Computational linguistics*, 42(2):163–205.
- Christos Christodouloupoulos and Mark Steedman. 2014. A massively parallel corpus: The Bible in 100 languages. *Language Resources and Evaluation*, pages 1–21.
- Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. [Unsupervised structure prediction with non-parallel multilingual guidance](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 50–61, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Michael Collins. 2002. [Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms](#). In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics.
- Joachim Daiber, Miloš Stanojević, and Khalil Sima'an. 2016. Universal reordering via linguistic typology. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3167–3176.
- Mona Diab and Steve Finch. 2000. A statistical word-level translation model for comparable corpora. In *Content-Based Multimedia Information Access-Volume 2*, pages 1500–1508. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.
- Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.
- Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. [Cross-lingual transfer for unsupervised dependency parsing without parallel data](#). In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 113–122, Beijing, China. Association for Computational Linguistics.
- Greg Durrett, Adam Pauls, and Dan Klein. 2012. [Syn-tactic transfer using a bilingual lexicon](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11, Jeju Island, Korea. Association for Computational Linguistics.
- Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 340–345. Association for Computational Linguistics.
- Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. [Dependency grammar induction via bitext projection constraints](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 369–377, Suntec, Singapore. Association for Computational Linguistics.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. [Cross-lingual dependency parsing based on distributed representations](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1234–1244, Beijing, China. Association for Computational Linguistics.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, Phoenix, Arizona, USA.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325.
- Diederik P. Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#). *CoRR*, abs/1412.6980.
- Ophélie Lacroix, Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. [Frustratingly easy cross-lingual transfer for transition-based dependency parsing](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1058–1063, San Diego, California. Association for Computational Linguistics.
- Xuezhe Ma and Fei Xia. 2014. [Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1:*

- Long Papers), pages 1337–1348, Baltimore, Maryland. Association for Computational Linguistics.
- Ryan McDonald, Slav Petrov, and Keith Hall. 2011. [Multi-source transfer of delexicalized dependency parsers](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.
- Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 629–637. Association for Computational Linguistics.
- Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.
- Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, et al. 2017. [Universal Dependencies 2](#). LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Robert Östling and Jörg Tiedemann. 2017. Neural machine translation for low-resource languages. *arXiv preprint arXiv:1708.05729*.
- Edoardo Maria Ponti, Roi Reichart, Anna Korhonen, and Ivan Vulić. 2018. Isomorphic transfer of syntactic structures in cross-lingual NLP. In *Proceedings of ACL*, Melbourne, Australia. Association for Computational Linguistics.
- Mohammad Sadegh Rasooli. 2019. *Cross-Lingual Transfer of Natural Language Processing Systems*. Ph.D. thesis, Columbia University.
- Mohammad Sadegh Rasooli and Michael Collins. 2015. [Density-driven cross-lingual transfer of dependency parsers](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 328–338, Lisbon, Portugal. Association for Computational Linguistics.
- Mohammad Sadegh Rasooli and Michael Collins. 2017. Cross-lingual syntactic transfer with limited resources. *Transactions of the Association of Computational Linguistics*, 5(1):279–293.
- Rudolf Rosa and Zdenek Zabokrtsky. 2015. [Klepos3 - a language similarity measure for delexicalized parser transfer](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 243–249, Beijing, China. Association for Computational Linguistics.
- Milan Straka and Jana Straková. 2017. [Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe](#). In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.
- Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. [Target language adaptation of discriminative transfer parsers](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia. Association for Computational Linguistics.
- Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 477–487. Association for Computational Linguistics.
- Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014. [Treebank translation for cross-lingual parser induction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 130–140, Ann Arbor, Michigan. Association for Computational Linguistics.
- Jörg Tiedemann and Željko Agić. 2016. Synthetic treebanking for cross-lingual dependency parsing. *Journal of Artificial Intelligence Research*, 55:209–248.
- Dingquan Wang and Jason Eisner. 2016. The galactic dependencies treebanks: Getting more data by synthesizing new languages. *Transactions of the Association for Computational Linguistics*, 4:491–505.
- Dingquan Wang and Jason Eisner. 2017. [Fine-grained prediction of syntactic typology: Discovering latent structure with supervised learning](#). *Transactions of the Association for Computational Linguistics*, 5:147–161.
- Dingquan Wang and Jason Eisner. 2018a. Surface statistics of an unknown language indicate how to parse it. *Transactions of the Association for Computational Linguistics*, 6:667–685.

- Dingquan Wang and Jason Eisner. 2018b. Synthetic data made to order: The case of parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. [Structured training for neural network transition-based parsing](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing, China. Association for Computational Linguistics.
- David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. [Inducing multilingual text analysis tools via robust projection across aligned corpora](#). In *Proceedings of the First International Conference on Human Language Technology Research, HLT '01*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.
- Yuan Zhang and Regina Barzilay. 2015. [Hierarchical low-rank tensors for multilingual transfer parsing](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1857–1867, Lisbon, Portugal. Association for Computational Linguistics.