Cross-Lingual Syntactic Transfer with Limited Resources

Mohammad Sadegh Rasooli and **Michael Collins***
Department of Computer Science, Columbia University
New York, NY 10027, USA
{rasooli,mcollins}@cs.columbia.edu

Abstract

We describe a simple but effective method for cross-lingual syntactic transfer of dependency parsers, in the scenario where a large amount of translation data is not available. This method makes use of three steps: 1) a method for deriving cross-lingual word clusters, which can then be used in a multilingual parser; 2) a method for transferring lexical information from a target language to source language treebanks; 3) a method for integrating these steps with the density-driven annotation projection method of Rasooli and Collins (2015). Experiments show improvements over the state-of-the-art in several languages used in previous work, in a setting where the only source of translation data is the Bible, a considerably smaller corpus than the Europarl corpus used in previous work. Results using the Europarl corpus as a source of translation data show additional improvements over the results of Rasooli and Collins (2015). We conclude with results on 38 datasets from the Universal Dependencies corpora.

1 Introduction

Creating manually-annotated syntactic treebanks is an expensive and time consuming task. Recently there has been a great deal of interest in cross-lingual syntactic transfer, where a parsing model is trained for some language of interest, using only treebanks in other languages. There is a clear motivation for this in building parsing models for languages for which treebank data is unavailable. Methods for syntactic transfer include annotation projection methods (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), learning of delexicalized models on universal treebanks (Zeman and Resnik, 2008; McDonald et al., 2011; Täckström et al., 2013; Rosa and Zabokrtsky, 2015), treebank translation (Tiedemann et al., 2014; Tiedemann, 2015; Tiedemann and Agić, 2016) and methods that leverage cross-lingual representations of word clusters, embeddings or dictionaries (Täckström et al., 2012; Durrett et al., 2012; Duong et al., 2015a; Zhang and Barzilay, 2015; Xiao and Guo, 2015; Guo et al., 2015; Guo et al., 2016; Ammar et al., 2016a).

This paper considers the problem of cross-lingual syntactic transfer with limited resources of monolingual and translation data. Specifically, we use the Bible corpus of Christodouloupoulos and Steedman (2014) as a source of translation data, and Wikipedia as a source of monolingual data. We deliberately limit ourselves to the use of Bible translation data because it is available for a very broad set of languages: the data from Christodouloupoulos and Steedman (2014) includes data from 100 languages. The Bible data contains a much smaller set of sentences (around 24,000) than other translation corpora, for example Europarl (Koehn, 2005), which has around 2 million sentences per language pair. This makes it a considerably more challenging corpus to work with. Similarly, our choice of Wikipedia as the source of monolingual data is motivated by the availability of Wikipedia data in a very broad set of languages.

*On leave at Google Inc. New York.

Transactions of the Association for Computational Linguistics, vol. 5, pp. 279–293, 2017. Action Editor: Yuji Matsumoto.
Submission batch: 5/2016; Revision batch: 10/2016; 2/2017; Published 8/2017.

We introduce a set of simple but effective methods for syntactic transfer, as follows:

- We describe a method for deriving cross-lingual clusters, where words from different languages with a similar syntactic or semantic role are grouped in the same cluster. These clusters can then be used as features in a shift-reduce dependency parser.
- We describe a method for transfer of lexical information from the target language into source language treebanks, using word-to-word translation dictionaries derived from parallel corpora. Lexical features from the target language can then be integrated in parsing.
- We describe a method that integrates the above two approaches with the density-driven approach to annotation projection described by Rasooli and Collins (2015).

Experiments show that our model outperforms previous work on a set of European languages from the Google universal treebank (McDonald et al., 2013). We achieve 80.9% average unlabeled attachment score (UAS) on these languages; in comparison the work of Zhang and Barzilay (2015), Guo et al. (2016) and Ammar et al. (2016b) have a UAS of 75.4%, 76.3% and 77.8%, respectively. All of these previous works make use of the much larger Europarl (Koehn, 2005) corpus to derive lexical representations. When using Europarl data instead of the Bible, our approach gives 83.9% accuracy, a 1.7% absolute improvement over Rasooli and Collins (2015). Finally, we conduct experiments on 38 datasets (26 languages) in the universal dependencies v1.3 (Nivre et al., 2016) corpus. Our method has an average unlabeled dependency accuracy of 74.8% for these languages, more than 6% higher than the method of Rasooli and Collins (2015). Thirteen datasets (10 languages) have accuracies higher than 80.0%.[1]

2 Background

This section gives a description of the underlying parsing models used in our experiments, the data sets used, and a baseline approach based on delexicalized parsing models.

2.1 The Parsing Model

We assume that the parsing model is a discriminative linear model, where given a sentence x, and a set of candidate parses $\mathcal{Y}(x)$, the output from the model is

$$y^*(x) = \arg \max_{y \in \mathcal{Y}(x)} \theta \cdot \phi(x, y)$$

where $\theta \in \mathbb{R}^d$ is a parameter vector, and $\phi(x, y)$ is a feature vector for the pair (x, y). In our experiments we use the shift-reduce dependency parser of Rasooli and Tetreault (2015), which is an extension of the approach in Zhang and Nivre (2011). The parser is trained using the averaged structured perceptron (Collins, 2002).

We assume that the feature vector $\phi(x, y)$ is the concatenation of three feature vectors:

- $\phi^{(p)}(x, y)$ is an unlexicalized set of features. Each such feature may depend on the part-of-speech (POS) tag of words in the sentence, but does not depend on the identity of individual words in the sentence.
- $\phi^{(c)}(x, y)$ is a set of cluster features. These features require access to a dictionary that maps each word in the sentence to an underlying cluster identity. Clusters may, for example, be learned using the Brown clustering algorithm (Brown et al., 1992). The features may make use of cluster identities in combination with POS tags.
- $\phi^{(l)}(x, y)$ is a set of lexicalized features. Each such feature may depend directly on word identities in the sentence. These features may also depend on part-of-speech tags or cluster information, in conjunction with lexical information.

Appendix A has a complete description of the features used in our experiments.

2.2 Data Assumptions

Throughout this paper we will assume that we have m source languages $\mathcal{L}_1 \ldots \mathcal{L}_m$, and a single target language $\mathcal{L}_{m+1}$. We assume the following data sources:

[1] The parser code is available at `https://github.com/rasoolims/YaraParser/tree/transfer`.

Source language treebanks. We have a treebank $\mathcal{T}_i$ for each language $i \in \{1 \ldots m\}$.

Part-of-speech (POS) data. We have hand-annotated POS data for all languages $\mathcal{L}_1 \ldots \mathcal{L}_{m+1}$. We assume that the data uses a universal POS set that is common across all languages.

Monolingual data. We have monolingual, raw text for each of the $(m+1)$ languages. We use $\mathcal{D}_i$ to refer to the monolingual data for the ith language.

Translation data. We have translation data for all language pairs. We use $\mathcal{B}_{i,j}$ to refer to translation data for the language pair (i, j) where $i, j \in \{1 \ldots (m+1)\}$ and $i \neq j$.

In our main experiments we use the Google universal treebank (McDonald et al., 2013) as our source language treebanks[2] (this treebank provides universal dependency relations and POS tags), Wikipedia data as our monolingual data, and the Bible from Christodouloupoulos and Steedman (2014) as the source of our translation data. In additional experiments we use the Europarl corpus as a source of translation data, in order to measure the impact of using the smaller Bible corpus.

2.3 A Baseline Approach: Delexicalized Parsers with Self-Training

Given the data assumption of a universal POS set, the feature vectors $\phi^{(p)}(x, y)$ can be shared across languages. A simple approach is then to simply train a delexicalized parser using treebanks $\mathcal{T}_1 \ldots \mathcal{T}_m$, using the representation $\phi(x, y) = \phi^{(p)}(x, y)$ (see (McDonald et al., 2013; Täckström et al., 2013)).

Our baseline approach makes use of a delexicalized parser, with two refinements:

WALS properties. We use the six properties from the World Atlas of Language Structures (WALS) (Dryer and Haspelmath, 2013) to select a subset of closely related languages for each target language. These properties are shown in Table 1. The model for a target language is trained on treebank data from languages where at least 4 out of 6 WALS properties are common between the source and target language.[3] This gives a slightly stronger baseline. Our experiments showed an improvement in average labeled dependency accuracy for the languages from 62.52% to 63.18%. Table 2 shows the set of source languages used for each target language. These source languages are used for all experiments in the paper.

Feature	Description
82A	Order of subject and verb
83A	Order of object and verb
85A	Order of adposition and noun phrase
86A	Order of genitive and noun
87A	Order of adjective and noun
88A	Order of demonstrative and noun

Table 1: The six properties from the world atlas of language structures (WALS) (Dryer and Haspelmath, 2013) used to select the source languages for each target language in our experiments.

Self-training. We use self-training (McClosky et al., 2006) to further improve parsing performance. Specifically, we first train a delexicalized model on treebanks $\mathcal{T}_1 \ldots \mathcal{T}_m$; then use the resulting model to parse a dataset $\mathcal{T}_{m+1}$ that includes target-language sentences which have POS tags but do not have dependency structures. We finally use the automatically parsed data $\mathcal{T}'_{m+1}$ as the treebank data and retrain the model. This last model is trained using all features (unlexicalized, clusters, and lexicalized). Self-training in this way gives an improvement in labeled accuracy from 63.18% to 63.91%.

2.4 Translation Dictionaries

Our only use of the translation data $\mathcal{B}_{i,j}$ for $i, j \in \{1 \ldots (m+1)\}$ is to construct a translation dictionary $t(w, i, j)$. Here i and j are two languages, w is a word in language $\mathcal{L}_i$, and the output $w' = t(w, i, j)$ is a word in language $\mathcal{L}_j$ corresponding to the most frequent translation of w into this language.

We define the function $t(w, i, j)$ as follows: We first run the GIZA++ alignment process (Och and Ney, 2003) on the data $\mathcal{B}_{i,j}$. We then keep intersected alignments between sentences in the two languages. Finally, for each word w in $\mathcal{L}_i$, we define

[2] We also train our best performing model on the newly released universal treebank v1.3 (Nivre et al., 2016). See §4.3 for more details.

[3] There was no effort to optimize this choice; future work may consider more sophisticated sharing schemes.

Target	Sources
en	de, fr, pt, sv
de	en, fr, pt
es	fr, it, pt
fr	en, de, es, it, pt, sv
it	es, fr, pt
pt	en, de, es, fr, it, sv
sv	en, fr, pt

Table 2: The selected source languages for each target language in the Google universal treebank v2 (McDonald et al., 2013). A language is chosen as a source language if it has at least 4 out of 6 WALS properties in common with the target language.

$w' = t(w, i, j)$ to be the target language word most frequently aligned to w in the aligned data. If a word w is never seen aligned to a target language word w', we define $t(w, i, j) = \texttt{NULL}$.

3 Our Approach

We now describe an approach that gives significant improvements over the baseline. §3.1 describes a method for deriving cross-lingual clusters, allowing us to add cluster features $\phi^{(c)}(x, y)$ to the model. §3.2 describes a method for adding lexical features $\phi^{(l)}(x, y)$ to the model. §3.3 describes a method for integrating the approach with the density-driven approach of Rasooli and Collins (2015). Finally, §4 describes experiments. We show that each of the above steps leads to improvements in accuracy.

3.1 Learning Cross-Lingual Clusters

We now describe a method for learning cross-lingual clusters. This follows previous work on cross-lingual clustering algorithms (Täckström et al., 2012). A *clustering* is a function $C(w)$ that maps each word w in a vocabulary to a cluster $C(w) \in \{1 \dots K\}$, where K is the number of clusters. A *hierarchical clustering* is a function $C(w, l)$ that maps a word w together with an integer l to a cluster at level l in the hierarchy. As one example, the Brown clustering algorithm (Brown et al., 1992) gives a hierarchical clustering. The level l allows cluster features at different levels of granularity.

A *cross-lingual* hierarchical clustering is a function $C(w, l)$ where the clusters are shared across the $(m + 1)$ languages of interest. That is, the word w can be from any of the $(m + 1)$ languages. Ideally, a cross-lingual clustering should put words across different languages which have a similar syntactic and/or semantic role in the same cluster. There is a clear motivation for cross-lingual clustering in the parsing context. We can use the cluster-based features $\phi^{(c)}(x, y)$ on the source language treebanks $\mathcal{T}_1 \dots \mathcal{T}_m$, and these features will now generalize beyond these treebanks to the target language $\mathcal{L}_{m+1}$.

Inputs: 1) Monolingual texts $\mathcal{D}_i$ for $i = 1 \dots (m + 1)$; 2) a function $t(w, i, j)$ that translates a word $w \in \mathcal{L}_i$ to $w' \in \mathcal{L}_j$; and 3) a parameter α such that $0 < \alpha < 1$.

Algorithm:

$\mathcal{D} = \{\}$
for i = 1 to $m + 1$ **do**
  for each sentence $s \in \mathcal{D}_i$ **do**
    for $p = 1$ to $|s|$ **do**
      Sample $\bar{a} \sim [0, 1)$
      if $\bar{a} \geq \alpha$ **then**
        continue
      Sample $j \sim unif\{1, ..., m + 1\} \backslash \{i\}$
      $w' = t(s_p, i, j)$
      if $w' \neq \texttt{NULL}$ **then**
        Set $s_p = w'$
    $\mathcal{D} = \mathcal{D} \cup \{s\}$
Use the algorithm of Stratos et al. (2015) on $\mathcal{D}$ to learn a clustering $\mathcal{C}$.

Output: The clustering $\mathcal{C}$.

Figure 1: An algorithm for learning a cross-lingual clustering. In our experiments we used the parameter value $\alpha = 0.3$.

We learn a cross-lingual clustering by leveraging the monolingual data sets $\mathcal{D}_1 \dots \mathcal{D}_{m+1}$, together with the translation dictionaries $t(w, i, j)$ learned from the translation data. Figure 1 shows the algorithm that learns a cross-lingual clustering. The algorithm first prepares a multilingual corpus, as follows: for each sentence s in the monolingual data $\mathcal{D}_i$, for each word in s, with probability α, we replace the word with its translation into some randomly chosen language. Once this data is created, we can easily obtain a cross-lingual clustering. Figure 1 shows the complete algorithm. The intuition behind this method is that by creating the cross-lingual data in this way, we bias the clustering al-

gorithm towards putting words that are translations of each other in the same cluster.

3.2 Treebank Lexicalization

We now describe how to introduce lexical representations $\phi^{(l)}(x, y)$ to the model. Our approach is simple: we take the treebank data $\mathcal{T}_1 \dots \mathcal{T}_m$ for the m source languages, together with the translation lexicons $t(w, i, m+1)$. For any word w in the source treebank data, we can look up its translation $t(w, i, m+1)$ in the lexicon, and add this translated form to the underlying sentence. Features can now consider lexical identities derived in this way. In many cases the resulting translation will be the `NULL` word, leading to the absence of lexical features. However, the representations $\phi^{(p)}(x, y)$ and $\phi^{(c)}(x, y)$ still apply in this case, so the model is robust to some words having a `NULL` translation.

3.3 Integration with the Density-Driven Projection Method of Rasooli and Collins (2015)

In this section we describe a method for integrating our approach with the cross-lingual transfer method of Rasooli and Collins (2015), which makes use of density-driven projections.

In annotation projection methods (Hwa et al., 2005; McDonald et al., 2011), it is assumed that we have translation data $\mathcal{B}_{i,j}$ for a source and target language, and that we have a dependency parser in the source language $\mathcal{L}_i$. The translation data consists of pairs (e, f) where e is a source language sentence, and f is a target language sentence. A method such as GIZA++ is used to derive an alignment between the words in e and f, for each sentence pair; the source language parser is used to parse e. Each dependency in e is then potentially transferred through the alignments to create a dependency in the target sentence f. Once dependencies have been transferred in this way, a dependency parser can be trained on the dependencies in the target language.

The density-driven approach of Rasooli and Collins (2015) makes use of various definitions of "density" of the projected dependencies. For example, $\mathcal{P}_{100}$ is the set of projected structures where the projected dependencies form a full projective parse tree for the sentence; $\mathcal{P}_{80}$ is the set of projected structures where at least 80% of the words in the projected structure are a modifier in some dependency. An iterative training process is used, where the parsing algorithm is first trained on the set $\mathcal{T}_{100}$ of complete structures, and where progressively less dense structures are introduced in learning.

We integrate our approach with the density-driven approach of Rasooli and Collins (2015) as follows: consider the treebanks $\mathcal{T}_1 \dots \mathcal{T}_m$ created using the lexicalization method of §3.2. We add all trees in these treebanks to the set $\mathcal{P}_{100}$ of full trees used to initialize the method of Rasooli and Collins (2015). In addition we make use of the representations $\phi^{(p)}$, $\phi^{(c)}$ and $\phi^{(l)}$, throughout the learning process.

4 Experiments

This section first describes the experimental settings, then reports results.

4.1 Data and Tools

Data In the first set of experiments, we consider 7 European languages studied in several pieces of previous work (Ma and Xia, 2014; Zhang and Barzilay, 2015; Guo et al., 2016; Ammar et al., 2016a; Lacroix et al., 2016). More specifically, we use the 7 European languages in the Google universal treebank (v.2; standard data) (McDonald et al., 2013). As in previous work, gold part-of-speech tags are used for evaluation. We use the concatenation of the treebank training sentences, Wikipedia data and the Bible monolingual sentences as our monolingual raw text. Table 3 shows statistics for the monolingual data. We use the Bible from Christodouloupoulos and Steedman (2014), which includes data for 100 languages, as the source of translations. We also conduct experiments with the Europarl data (both with the original set and a subset of it with the same size as the Bible) to study the effects of translation data size and domain shift. The statistics for translation data is shown in Table 4.

In a second set of experiments, we run experiments on 38 datasets (26 languages) in the more recent Universal Dependencies v1.3 corpus (Nivre et al., 2016). The full set of languages we use is listed in Table 9.[4] We use the Bible as the translation data,

[4]We excluded languages that are not completely present in the Bible of Christodouloupoulos and Steedman (2014) (An-

and Wikipedia as the monolingual text. The standard training, development and test set splits are used in all experiments. The development sets are used for analysis, given in §5 of this paper.

Lang.	en	de	es	fr	it	pt	sv
#Sen.	31.8	20.0	13.6	13.6	10.1	6.1	3.9
#Token	750.5	408.2	402.3	372.1	311.1	169.3	60.6
#Type	3.8	6.1	2.7	2.4	2.1	1.6	1.3

Table 3: Sizes of the monolingual datasets for each of our languages. All numbers are in millions.

Brown Clustering Algorithm We use the off-the-shelf Brown clustering tool[5] (Liang, 2005) to train monolingual Brown clusters with 500 clusters. The monolingual Brown clusters are used as features over lexicalized values created in $\phi^{(l)}$, and in self-training experiments. We train our cross-lingual clustering with the off-the-shelf-tool[6] from Stratos et al. (2015). We set the window size to 2 with a cluster size of 500.[7]

Parsing Model We use the k-beam arc-eager dependency parser of Rasooli and Tetreault (2015), which is similar to the model of Zhang and Nivre (2011). We modify the parser such that it can use both monolingual and cross-lingual word cluster features. The parser is trained using the the maximum violation update strategy (Huang et al., 2012). We use three epochs of training for all experiments. We use the DEPENDABLE Tool (Choi et al., 2015) to calculate significance tests on several of the comparisons (details are given in the captions to tables 5, 6, and 9).

cient Greek, Basque, Catalan, Galician, Gothic, Irish, Kazakh, Latvian, Old Church Slavonic, and Tamil). We also excluded Arabic, Hebrew, Japanese and Chinese, as these languages have tokenization and/or morphological complexity that goes beyond the scope of this paper. Future work should consider these languages.

[5] https://github.com/percyliang/brown-cluster

[6] https://github.com/karlstratos/singular

[7] Usually the original Brown clusters are better features for parsing but their training procedure does not scale well to large datasets. Therefore we use the more efficient algorithm from Stratos et al. (2015) on the larger cross-lingual datasets to obtain word clusters.

Data	Lang.	en	de	es	fr	it	pt	sv
Bible	tokens	1.5M	665K	657K	732K	613K	670K	696K
	types	16K	20K	27K	22K	29K	29K	23K
EU-S	tokens	718K	686K	753K	799K	717K	739K	645K
	types	22K	41K	31K	27K	30K	32K	39K
Europarl	tokens	56M	50M	57M	62M	55M	56M	46M
	types	133K	400K	195K	153K	188K	200K	366K

Table 4: Statistics for the Bible, sampled Europarl (EU-S) and Europarl datasets. Each individual Bible text file from Christodouloupoulos and Steedman (2014) consists of 24720 sentences, except for English datasets, where two translations into English are available, giving double the amount of data. Each text file from the sampled Europarl datasets consists of 25K sentences and Europarl has approximately 2 million sentences per language pair.

L	Baseline		This paper using the Bible					
			§3.1		§3.2		§3.3	
	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS
en	58.2	65.5	65.0	72.3	66.3	74.0	**70.8**	**76.5**
de	49.7	59.1	51.6	59.7	54.9	62.6	**65.2**	**72.8**
es	68.3	77.2	73.1	79.6	76.6	81.9	**76.7**	**82.1**
fr	67.3	77.7	69.5	79.9	74.4	81.9	**75.8**	**82.2**
it	69.7	79.4	71.6	80.0	74.7	82.8	**76.1**	**83.3**
pt	71.5	77.5	76.9	81.5	81.0	84.4	**81.3**	**84.7**
sv	62.6	74.2	63.5	75.1	68.2	78.7	**71.2**	**80.3**
avg	63.9	72.9	67.3	75.5	70.9	78.1	**73.9**	**80.3**

Table 5: Performance of different models in this paper; first the baseline model, then models trained using the methods described in sections §3.1–3.3. All results make use of the Bible as a source of translation data. All differences in UAS and LAS are statistically significant with $p < 0.001$ using McNemar's test, with the exception of "de" UAS/LAS Baseline vs. 3.1 (i.e., 49.7 vs 51.6 UAS and 59.1 vs 59.7 LAS are not significant differences).

Word alignment We use the intersected alignments from GIZA++ (Och and Ney, 2003) on translation data. We exclude sentences in translation data with more than 100 words.

4.2 Results on the Google Treebank

Table 5 shows the dependency parsing accuracy for the baseline delexicalized approach, and for models which add 1) cross-lingual clusters (§3.1); 2) lexical features (§3.2); and 3) integration with the density-driven method of Rasooli and Collins (2015). Each of these three steps gives significant improvements in performance. The final LAS/UAS of 73.9/80.3% is several percentage points higher than the baseline accuracy of 63.9/72.9%.

Lang.	Bible				Europarl-Sample				Europarl			
	Density		This Paper		Density		This Paper		Density		This Paper	
	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS
en	59.1	66.4	70.8	76.5	64.3	72.8	70.2	76.2	68.4	76.3	71.1	77.5
de	60.2	69.5	65.2	72.8	61.6	72.0	64.9	73.0	73.0	79.7	75.6	82.1
es	70.3	76.8	76.7	82.1	72.0	78.3	76.0	81.5	74.6	80.9	76.6	82.6
fr	69.9	76.9	75.8	82.2	71.9	79.0	75.7	82.5	76.3	82.7	77.4	83.9
it	71.1	78.5	76.1	83.3	73.2	80.4	76.2	82.9	77.0	83.7	77.4	84.4
pt	72.1	76.4	81.3	84.7	75.3	79.7	81.61	84.8	77.3	82.1	82.1	85.6
sv	66.5	76.3	71.2	80.3	71.9	80.6	73.5	81.6	75.6	84.1	76.9	84.5
avg	67.0	75.7	73.9	80.3	70.0	77.6	74.0	80.4	74.6	81.3	76.7	82.9

Table 6: Results for our method using different sources of translation data. "Density" refers to the method of Rasooli and Collins (2015); "This paper" gives results using the methods described in sections 3.1–3.3 of this paper. The "Bible" experiments use the Bible data of Christodouloupoulos and Steedman (2014). The "Europarl" experiments use the Europarl data of Koehn (2005). The "Europarl-Sample" experiments use 25K randomly chosen sentences from Europarl; this gives a similar number of sentences to the Bible data. All differences in LAS and UAS in this table between the density and "this paper" settings (i.e., for the Bible, Europarl-Sample and Europarl settings) are found to be statistically significant according to McNemar's sign test.

Lang.	MX14	LA16	ZB15		GCY16		AMB16		RC15		This paper				Supervised	
											Bible		Europarl			
	UAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS		
en	–	–	59.8	70.5	–	–	–	–	68.4	76.3	70.8	76.5	**71.1**	**77.5**	92.0	93.8
de	74.3	76.0	54.1	62.5	55.9	65.0	57.1	65.2	73.0	79.7	65.2	72.8	**75.6**	**82.1**	79.4	85.3
es	75.5	78.9	68.3	78.0	73.0	79.0	74.6	80.2	74.6	80.9	**76.7**	82.1	76.0	**82.6**	82.3	86.7
fr	76.5	80.8	68.8	78.9	71.0	77.6	73.9	80.6	76.3	82.7	75.8	82.2	**77.4**	**83.9**	81.7	86.3
it	77.7	79.4	69.4	79.3	71.2	78.4	72.5	80.7	77.0	83.7	76.1	83.3	**77.4**	**84.4**	86.1	88.8
pt	76.6	–	72.5	78.6	78.6	81.8	77.0	81.2	77.3	82.1	81.3	84.7	**82.1**	**85.6**	87.6	89.4
sv	79.3	83.0	62.5	75.0	69.5	78.2	68.1	79.0	75.6	84.1	71.2	80.3	**76.9**	**84.5**	84.1	88.1
$avg_{\backslash en}$	76.7	–	65.9	75.4	69.3	76.3	70.5	77.8	75.6	82.2	74.4	80.9	**77.7**	**83.9**	83.5	87.4

Table 7: Comparison of our work using the Bible and Europarl data, with previous work: MX14 (Ma and Xia, 2014), LA16 (Lacroix et al., 2016), ZB15 (Zhang and Barzilay, 2015), GCY16 (Guo et al., 2016), AMB 16 (Ammar et al., 2016b), and RC15 (Rasooli and Collins, 2015). "Supervised" refers to the performance of the parser trained on fully gold standard data in a supervised fashion (i.e. the practical upper-bound of our model). "$avg_{\backslash en}$" refers to the average accuracy for all datasets except English.

Comparison to the Density-Driven Approach using Europarl Data Table 6 shows accuracies for the density-driven approach of Rasooli and Collins (2015), first using Europarl data[8] and second using the Bible alone (with no cross-lingual clusters or lexicalization). The Bible data is considerably smaller than Europarl (around 100 times smaller), and it can be seen that results using the Bible are several percentage points lower than the results for Europarl (75.7% UAS vs. 81.3% UAS). Integrating cluster-based and lexicalized features described in the current paper with the density-driven approach closes much of this gap in performance (80.3% UAS). Thus we have demonstrated that we can get close to the performance of the Europarl-based models using only the Bible as a source of translation data. Using our approach on the full Europarl data gives an average UAS of 82.9%, an improvement from the 81.3% UAS of Rasooli and Collins (2015).

Table 6 also shows results when we use a random subset of the Europarl data, in which the number of sentences (25,000) is chosen to give a very similar size to the Bible. It can be seen that accuracies using the Bible vs. the Europarl-Sample are very similar (80.3% vs. 80.4% UAS), suggesting that the size of the translation corpus is much more important than the genre.

Comparison to Other Previous Work Table 7 compares the accuracy of our method to the following related work: 1) Ma and Xia (2014), who describe an annotation projection method based on entropy regularization; 2) Lacroix et al. (2016), who

[8]Rasooli and Collins (2015) do not report results on English. We use the same setting to obtain the English results.

Lang.	RC15		This Paper (§3.3)			
			Bible		Europarl	
	LAS	UAS	LAS	UAS	LAS	UAS
en	66.2	74.4	67.8	74.4	**68.0**	**75.1**
de	71.6	78.8	61.9	70.3	**73.6**	**80.8**
es	72.3	79.2	73.8	79.9	**74.2**	**80.7**
fr	73.5	80.8	72.6	79.9	**75.0**	**82.3**
it	74.9	82.0	74.0	81.7	**75.3**	**82.6**
pt	75.4	80.7	79.2	83.3	**80.4**	**84.4**
sv	73.4	82.0	67.3	77.2	**73.7**	**82.2**
avg	72.5	79.7	70.9	78.1	**74.3**	**81.2**

Table 8: The final results based on automatic part of speech tags. RC15 refers to the best performing model of Rasooli and Collins (2015).

describe an annotation projection method based on training on partial trees with dynamic oracles; 3) Zhang and Barzilay (2015), who describe a method that learns cross-lingual embeddings and bilingual dictionaries from Europarl data, and uses these features in a discriminative parsing model; 4) Guo et al. (2016), who describe a method that learns cross-lingual embeddings from Europarl data and uses a shift-reduce neural parser with these representations; 5) Ammar et al. (2016b)[9], who use the same embeddings as Guo et al. (2016), within an LSTM-based parser; and 6) Rasooli and Collins (2015) who use the density-driven approach on the Europarl data. Our method gives significant improvements over the first three models, in spite of using the Bible translation data rather than Europarl. When using the Europarl data, our method improves the state-of-the-art model of Rasooli and Collins (2015).

Performance with Automatic POS Tags For completeness, Table 8 gives results for our method with automatic part-of-speech tags. The tags are obtained using the model of Collins (2002)[10] trained on the training part of the treebank dataset. Future work should study approaches that transfer POS tags in addition to dependencies.

4.3 Results on the Universal Dependencies v1.3

Table 9 gives results on 38 datasets (26 languages) from the newly released universal dependencies corpus (Nivre et al., 2016). Given the number of treebanks and to speed up training, we pick source languages that have at least 5 out of 6 common WALS properties with each target language. Our experiments are carried out using the Bible as our transla-

Dataset	Density		This paper		Supervised	
	LAS	UAS	LAS	UAS	LAS	UAS
it	74.3	81.3	79.8	86.1	88.4	90.7
sl	68.2	75.9	78.6	84.1	86.3	89.1
es	69.1	77.5	76.3	84.1	83.5	86.9
bg	66.2	79.5	72.0	83.6	85.5	90.5
pt	66.7	75.8	74.8	83.4	83.0	86.7
es-ancora	68.9	77.5	74.6	83.1	86.5	89.4
fr	72.0	77.9	76.6	82.6	84.5	87.1
sv-lines	67.5	76.7	73.3	82.4	81.0	85.4
pt-br	68.3	75.2	76.2	82.0	87.8	89.7
sv	65.9	75.7	71.7	81.3	83.6	87.7
no	71.7	78.8	74.3	81.2	88.0	90.5
pl	65.4	77.6	70.1	81.0	85.1	90.3
hr	55.8	70.2	65.9	80.9	76.2	85.1
cs-cac	61.1	70.3	69.0	78.5	82.4	87.6
da	63.1	72.8	68.3	77.8	80.8	84.3
en-lines	67.0	75.9	68.6	77.3	80.7	84.6
cs	59.0	68.1	67.2	76.4	84.5	88.7
id	38.0	55.7	57.8	76.0	79.8	85.1
de	61.3	72.8	64.9	75.7	80.2	85.8
ru-syntagrus	56.0	70.7	61.6	75.3	82.0	87.8
ru	56.7	64.8	65.4	74.8	71.9	77.7
cs-cltt	57.5	65.4	65.6	74.7	77.1	81.4
ro	54.6	67.4	60.7	74.6	78.2	85.3
la	54.5	71.6	55.7	72.8	43.1	52.5
nl-lassysmall	51.5	62.6	61.9	71.7	76.5	80.6
el	53.7	66.7	59.6	71.0	79.1	83.1
et	48.9	65.6	56.9	70.9	75.9	82.9
hi	34.4	50.6	49.9	69.9	89.4	92.9
hu	26.1	48.9	55.0	69.9	69.5	79.4
en	59.7	68.1	61.8	69.0	85.3	88.1
fi-ftb	50.3	63.2	56.5	67.5	73.3	79.7
fi	49.8	60.8	57.3	66.4	73.4	78.2
la-ittb	44.1	55.4	51.8	62.8	76.2	80.9
nl	40.6	49.4	50.1	62.0	70.1	75.0
la-proiel	43.6	60.3	45.0	61.3	64.9	72.9
sl-sst	42.4	59.2	47.6	60.6	63.4	70.4
fa	44.4	53.2	46.5	56.0	84.1	87.5
tr	05.3	18.5	32.7	51.9	65.6	78.8
Average	56.7	68.1	64.0	74.8	78.9	83.8

Table 9: Results for the density driven method (Rasooli and Collins, 2015) and ours using the Bible data on the universal dependencies v1.3 (Nivre et al., 2016). The table is sorted by the performance of our method. The last major columns shows the performance of the supervised parser. The abbreviations are as follows: bg (Bulgarian), cs (Czech), da (Danish), de (German), el (Greek), en (English), es (Spanish), et (Estonian), fa (Persian (Farsi)), fi (Finnish), fr (French), hi (Hindi), hr (Croatian), hu (Hungarian), id (Indonesian), it (Italian), la (Latin), nl (Dutch), no (Norwegian), pl (Polish), pt (Portuguese), ro (Romanian), ru (Russian), sl (Slovenian), sv (Swedish), and tr (Turkish). All differences in LAS and UAS in this table were found to be statistically significant according to McNemar's sign test with $p < 0.001$.

[9]This work was later published under a different title (Ammar et al., 2016a) without including UAS results.

[10]https://github.com/rasoolims/SemiSupervisedPosTagger

tion data. As shown in Table 9, our method consistently outperforms the density-driven method of Rasooli and Collins (2015) and for many languages the accuracy of our method gets close to the accuracy of the supervised parser. In all the languages, our method is significantly better than the density-driven method using the McNemar's test with $p < 0.001$.

Accuracy on some languages (e.g., Persian (fa) and Turkish (tr)) is low, suggesting that future work should consider more powerful techniques for these languages. There are two important facts to note. First, the number of fully projected trees in some languages is so low such that the density-driven approach cannot start with a good initialization to fill in partial dependencies. For example Turkish has only one full tree with only six words, Persian with 25 trees, and Dutch with 28 trees. Second, we observe very low accuracies in supervised parsing for some languages in which the number of training sentences is very low (for example, Latin has only 1326 projective trees in the training data).

5 Analysis

We conclude with some analysis of the accuracy of the method on different dependency types, across the different languages. Table 10 shows precision and recall on different dependency types in English (using the Google treebank). The improvements in accuracy when moving from the delexicalized model to the Bible or Europarl model apply quite uniformly across all dependency types, with all dependency labels showing an improvement.

Table 11 shows the dependency accuracy sorted by part-of-speech tag of the modifier in the dependency. We break the results into three groups: G1 languages, where UAS is at least 80% overall; G2 languages, where UAS is between 70% and 80%; and G3 languages, where UAS is less than 70%. There are some quite significant differences in accuracy depending on the POS of the modifier word. In the G1 languages, for example, ADP, DET, ADJ, PRON and AUX all have over 85% accuracy; in contrast NOUN, VERB, PROPN, ADV all have accuracy that is less than 80%. A very similar pattern is seen for the G2 languages, with ADP, DET, ADJ, and AUX again having greater than 85% accuracy, but NOUN, VERB, PROPN and ADV having lower accuracies. These results suggest that difficulty varies quite significantly depending on the modifier POS, and different languages show the same patterns of difficulty with respect to the modifier POS.

Table 12 shows accuracy sorted by the POS tag of the *head* word of the dependency. By far the most frequent head POS tags are NOUN, VERB, and PROPN (accounting for 85% of all dependencies). The table also shows that for all language groups G1, G2, and G3, the f1 scores for NOUN, VERB and PROPN are generally higher than the f1 scores for other head POS tags.

Finally, Table 13 shows precision and recall for different dependency labels for the G1, G2 and G3 languages. We again see quite large differences in accuracy between different dependency labels. The G1 language dependencies, with the most frequent label `nmod`, has an F-score of 75.2. In contrast, the second most frequent label, `case`, has 93.7 F-score. Other frequent labels with low accuracy in the G1 languages are `advmod`, `conj`, and `cc`.

6 Related Work

There has recently been a great deal of work on syntactic transfer. A number of methods (Zeman and Resnik, 2008; McDonald et al., 2011; Cohen et al., 2011; Naseem et al., 2012; Täckström et al., 2013; Rosa and Zabokrtsky, 2015) directly learn delexicalized models that can be trained on universal treebank data from one or more source languages, then applied to the target language. More recent work has introduced cross-lingual representations—for example cross-lingual word-embeddings—that can be used to improve performance (Zhang and Barzilay, 2015; Guo et al., 2015; Duong et al., 2015a; Duong et al., 2015b; Guo et al., 2016; Ammar et al., 2016b). These cross-lingual representations are usually learned from parallel translation data. We show results of several methods (Zhang and Barzilay, 2015; Guo et al., 2016; Ammar et al., 2016b) in Table 7 of this paper.

The annotation projection approach, where dependencies from one language are transferred through translation alignments to another language, has been considered by several authors (Hwa et al., 2005; Ganchev et al., 2009; McDonald et al., 2011; Ma and Xia, 2014; Rasooli and Collins, 2015;

dependency	freq	Delexicalized		Bible		Europarl	
		prec./rec.	f1	prec./rec.	f1	prec./rec.	f1
adpmod	10.6	57.2/62.7	59.8	67.1/71.8	69.4	70.3/73.8	72.0
adpobj	10.6	65.5/69.1	67.2	75.3/77.4	76.3	75.9/79.2	77.6
det	9.5	72.5/75.6	74.0	84.3/86.3	85.3	86.6/89.8	88.2
compmod	9.1	83.7/ 59.9	69.8	87.3/70.2	77.8	89.0/73.0	80.2
nsubj	8.0	69.7/60.0	64.5	82.1/77.5	79.7	83.0/78.1	80.5
amod	7.0	76.9/72.3	74.5	83.0/78.7	80.8	80.9/77.9	79.4
ROOT	4.8	69.3/70.4	69.8	85.0/85.1	85.0	83.8/85.8	84.8
num	4.6	67.8/55.3	60.9	70.7/55.2	62.0	75.0/63.0	68.5
dobj	4.5	60.8/80.3	69.2	64.0/84.9	73.0	68.4/86.6	76.5
advmod	4.1	65.9/61.9	63.8	72.7/68.1	70.3	69.6/68.8	69.2
aux	3.5	76.6/93.9	84.4	90.2/95.9	93.0	89.6/96.4	92.9
cc	2.9	67.6/61.7	64.5	73.1/73.1	73.1	73.1/73.3	73.2
conj	2.8	46.3/56.1	50.7	45.6/62.9	52.9	48.1/62.8	54.5
dep	2.0	90.5/25.8	40.1	99.2/33.8	50.4	92.0/34.4	50.1
poss	2.0	72.1/30.6	43.0	77.9/45.8	57.7	78.2/42.1	54.7
ccomp	1.6	76.2/28.4	41.3	88.0/61.3	72.3	82.3/69.1	75.1
adp	1.2	20.0/0.5	0.9	92.7/42.1	57.9	91.7/23.3	37.1
nmod	1.2	60.7/48.1	53.7	56.3/47.1	51.3	52.6/46.2	49.2
xcomp	1.2	66.6/48.6	56.2	85.1/65.3	73.9	78.3/71.0	74.5
mark	1.1	37.8/24.6	29.8	73.8/50.3	59.8	62.8/53.8	57.9
advcl	0.8	23.6/22.3	22.9	38.7/38.8	38.8	38.0/42.9	40.3
appos	0.8	8.5/43.0	14.3	20.4/61.0	30.6	26.4/61.7	37.0
auxpass	0.8	88.9/91.4	90.1	96.8/97.1	97.0	98.6/98.6	98.6
rcmod	0.8	38.2/33.3	35.6	46.8/54.6	50.4	52.7/55.0	53.8
nsubjpass	0.7	73.2/64.9	68.8	87.6/77.0	82.0	85.5/75.8	80.3
acomp	0.6	86.8/92.5	89.6	83.3/93.5	88.1	91.0/93.9	92.4
adpcomp	0.6	42.0/70.2	52.5	47.9/61.5	53.9	55.4/47.1	50.9
partmod	0.6	20.2/36.0	25.8	36.7/49.1	42.0	31.0/40.7	35.2
attr	0.5	67.7/86.4	75.9	76.5/92.1	83.6	72.6/92.7	81.4
neg	0.5	74.7/85.0	79.6	93.3/91.0	92.1	92.6/89.8	91.2
prt	0.3	27.4/92.2	42.2	32.4/96.6	48.5	31.9/97.4	48.1
infmod	0.2	30.7/72.4	43.2	38.4/64.4	48.1	42.6/63.2	50.9
expl	0.1	84.8/87.5	86.2	93.8/93.8	93.8	91.2/96.9	93.9
iobj	0.1	51.7/78.9	62.5	88.9/84.2	86.5	36.4/84.2	50.8
mwe	0.1	0.0/0.0	0.0	5.3/2.1	3.0	11.1/10.4	10.8
parataxis	0.1	5.6/19.6	8.7	17.3/47.1	25.3	14.6/45.1	22.0
cop	0.0	0.0/0.0	0.0	0.0/0.0	0.0	0.0/0.0	0.0
csubj	0.0	12.8/33.3	18.5	22.2/26.7	24.2	25.0/46.7	32.6
csubjpass	0.0	100.0/100.0	100.0	100.0/100.0	100.0	50.0/100.0	66.7
rel	0.0	100.0/6.3	11.8	90.9/62.5	74.1	66.7/37.5	48.0

Table 10: Precision, recall and f-score of different dependency relations on the English development data of the Google universal treebank. The major columns show the dependency labels ("dep."), frequency ("freq."), the baseline delexicalized model ("delex"), and our method using the Bible and Europarl ("EU") as translation data. The rows are sorted by frequency.

POS	G1		G2		G3	
	freq%	acc.	freq%	acc.	freq%	acc.
NOUN	22.0	77.6	30.0	71.2	25.3	58.0
ADP	16.9	92.3	10.9	92.3	11.2	90.6
DET	11.9	96.4	3.0	92.4	3.6	86.6
VERB	11.7	74.5	13.5	66.1	17.1	52.2
PROPN	8.1	79.0	4.7	65.2	6.8	49.5
ADJ	8.0	88.5	12.7	86.9	8.4	73.6
PRON	5.4	87.7	5.9	82.2	7.6	71.1
ADV	4.3	76.0	6.6	70.9	5.6	61.9
CONJ	3.6	71.8	4.7	63.0	4.2	60.4
AUX	2.7	91.5	1.7	88.9	3.0	70.6
NUM	2.2	79.5	2.3	68.4	2.0	75.7
SCONJ	1.8	80.5	1.9	77.2	2.6	65.0
PART	0.9	80.2	1.8	64.3	1.9	45.0
X	0.2	52.3	0.1	40.5	0.6	36.9
SYM	0.1	64.3	0.1	40.9	0.1	45.5
INTJ	0.1	78.5	0.0	51.7	0.3	60.2

Table 11: Accuracy of unlabeled dependencies by POS of the modifier word, for three groups of languages for the universal dependencies experiments in Table 9: G1 (languages with UAS $\geq$ 80), G2 (languages with 70 $\leq$ UAS $<$ 80), G3 (languages with UAS $<$ 70). The rows are sorted by frequency in the G1 languages.

Lacroix et al., 2016; Agić et al., 2016; Schlichtkrull and Søgaard, 2017).

Other recent work (Tiedemann et al., 2014; Tiedemann, 2015; Tiedemann and Agić, 2016) has considered treebank translation, where a statistical machine translation system (e.g., MOSES (Koehn et al., 2007)) is used to translate a source language treebank into the target language, complete with reordering of the input sentence. The lexicalization approach described in this paper is a simple form of treebank translation, where we use a word-to-word translation model. In spite of its simplicity, it is an effective approach.

A number of authors have considered incorporating universal syntactic properties, such as dependency order, by selectively learning syntactic attributes from similar source languages (Naseem et al., 2012; Täckström et al., 2013; Zhang and Barzilay, 2015; Ammar et al., 2016a). Selective sharing of syntactic properties is complementary to our work. We used a very limited form of selective sharing, through the WALS properties, in our baseline approach. More recently, Wang and Eisner (2016) have developed a synthetic treebank as a universal treebank to help learn parsers for new languages. Martínez Alonso et al. (2017) try a very different approach in cross-lingual transfer by using a ranking approach.

A number of authors (Täckström et al., 2012; Guo et al., 2015; Guo et al., 2016) have introduced methods that learn cross-lingual representations that are then used in syntactic transfer. Most of these approaches introduce constraints to a clustering or embedding algorithm that encourage words that are translations of each other to have similar representations. Our method of deriving a cross-lingual cor-

POS	G1				G2				G3			
	freq%	prec.	rec.	f1	freq%	prec.	rec.	f1	freq%	prec.	rec.	f1
NOUN	43.9	85.4	88.6	87.0	43.5	77.3	81.2	79.2	34.5	67.1	71.0	69.0
VERB	32.0	83.5	83.6	83.6	35.4	74.9	77.9	76.4	41.3	63.8	66.5	65.1
PROPN	9.1	84.0	84.0	84.0	4.1	67.6	63.2	65.3	6.4	57.2	54.8	56.0
ADJ	4.5	76.2	72.4	74.3	5.7	75.7	56.0	64.4	5.8	64.9	49.1	55.9
PRON	1.4	79.3	68.3	73.4	1.4	81.5	61.4	70.0	2.2	65.2	49.1	56.0
NUM	1.2	77.2	72.4	74.7	1.0	52.0	41.8	46.3	0.7	62.5	54.7	58.3
ADV	1.0	54.0	39.0	45.3	1.5	56.5	27.2	36.7	1.2	44.1	25.8	32.6
ADP	0.6	39.8	6.5	11.2	0.3	25.0	0.9	1.7	0.3	40.5	8.3	13.8
SYM	0.3	79.0	81.1	80.1	0.1	41.5	66.3	51.0	0.1	55.3	52.2	53.7
DET	0.3	36.3	22.6	27.8	0.1	60.6	30.6	40.7	0.1	67.6	25.3	36.8
AUX	0.2	35.7	3.7	6.6	0.0	17.2	6.7	9.6	0.8	33.3	2.2	4.2
X	0.1	52.4	52.2	52.3	0.1	42.5	41.6	42.1	0.4	39.7	42.7	41.1
SCONJ	0.1	36.8	10.0	15.7	0.1	45.7	5.8	10.3	0.1	30.0	13.5	18.7
PART	0.1	26.7	3.0	5.4	0.1	15.9	4.3	6.8	0.1	26.7	36.8	30.9
CONJ	0.1	47.8	6.5	11.4	0.1	3.3	0.9	1.4	0.1	51.7	10.2	17.0
INTJ	0.0	52.4	47.8	50.0	0.0	20.0	7.1	10.5	0.1	44.2	43.0	43.6

Table 12: Precision, recall and f-score of unlabeled dependency attachment for different POS tags *as head* for three groups of languages for the universal dependencies experiments in Table 9: G1 (languages with UAS $\geq$ 80), G2 (languages with 70 $\leq$ UAS $<$ 80), G3 (languages with UAS $<$ 70). The rows are sorted by frequency in the G1 languages.

pus (see Figure 1) is closely related to Duong et al. (2015a); Gouws and Søgaard (2015); and Wick et al. (2015).

Our work has made use of dictionaries that are automatically extracted from bilingual corpora. An alternative approach would be to use hand-crafted translation lexicons, for example, PanLex (Baldwin et al., 2010) (e.g. see Duong et al. (2015b)), which covers 1253 language varieties, Google translate (e.g., see Ammar et al. (2016c)), or Wiktionary (e.g., see Durrett et al. (2012) for an approach that uses Wiktionary for cross-lingual transfer). These resources are potentially very rich sources of information. Future work should investigate whether they can give improvements in performance.

7 Conclusions

We have described a method for cross-lingual syntactic transfer that is effective in a scenario where a large amount of translation data is not available. We have introduced a simple, direct method for deriving cross-lingual clusters, and for transferring lexical information across treebanks for different languages. Experiments with this method show that the method gives improved performance over previous work that makes use of Europarl, a much larger translation corpus.

Acknowledgement

We thank the anonymous reviewers for their valuable feedback. We also thank Ryan McDonald, Karl Stratos and Oscar Täckström for their comments on the first draft.

Appendix A Parsing Features

We used all features in Zhang and Nivre (2011, Table 1 and 2), which describes features based on the word and part-of-speech at various positions on the stack and buffer of the transition system. In addition, we expand the Zhang and Nivre (2011, Table 1) features to include clusters, as follows: whenever a feature tests the part-of-speech for a word in position 0 of the stack or buffer, we introduce features that replace the part-of-speech with the Brown clustering bit-string of length 4 and 6. Whenever a feature tests for the word identity at position 0 of the stack or buffer, we introduce a cluster feature that replaces the word with the full cluster feature. We take the cross product of all features corresponding to the choice of 4 or 6 length bit string for part-of-speech features.

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Alonso Martínez, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for

Dep.	G1				G2				G3			
	freq%	prec.	rec.	f1	freq%	prec.	rec.	f1	freq%	prec.	rec.	f1
nmod	15.8	74.0	76.3	75.2	16.4	67.3	72.2	69.7	17.3	56.9	57.6	57.3
case	15.3	92.6	94.7	93.7	10.7	92.4	93.5	93.0	10.7	90.2	90.2	90.2
det	11.8	96.5	96.4	96.4	3.5	91.8	91.9	91.9	3.8	79.1	86.4	82.6
nsubj	6.5	85.3	86.8	86.0	7.5	75.5	73.5	74.5	7.8	61.0	63.2	62.1
amod	6.4	92.9	94.0	93.5	10.8	90.1	90.9	90.5	5.3	75.7	82.9	79.1
dobj	5.3	93.0	90.8	91.9	7.1	84.3	81.8	83.0	5.7	71.9	72.6	72.3
root	5.3	84.8	85.2	85.0	6.8	77.5	77.9	77.7	7.9	64.9	65.7	65.3
advmod	4.1	73.4	72.2	72.8	7.1	68.1	69.3	68.7	5.3	54.8	58.7	56.7
conj	4.0	60.4	68.1	64.0	5.8	50.2	56.6	53.2	4.2	41.3	48.1	44.5
cc	3.4	71.2	71.2	71.2	4.5	63.5	63.3	63.4	3.9	60.6	61.6	61.1
mark	3.3	85.1	87.0	86.0	2.2	76.2	79.6	77.9	3.4	70.9	71	71
acl	2.4	65.9	61.6	63.7	1.7	49.7	51.3	50.5	2.0	32.6	28.7	30.5
aux	2.2	91.5	93.6	92.5	1.2	86.8	91.1	88.9	2.2	66.4	78.2	71.8
name	1.9	86.5	86.2	86.4	1.3	75.3	72.1	73.6	0.8	27.8	45.1	34.4
cop	1.6	73.1	74.5	73.8	1.3	67.7	52.5	59.1	2.1	50.8	51.2	51
nummod	1.4	83.8	86.0	84.9	1.6	73.9	77.6	75.7	1.4	79.2	81.7	80.5
advcl	1.3	60.1	59.8	60.0	1.3	57.4	48.8	52.7	2.0	42.6	38.1	40.2
appos	1.3	73.9	64.9	69.1	0.8	51.2	48.9	50.0	0.5	31.3	32.1	31.7
mwe	0.9	57.7	15.6	24.6	0.5	66.2	15.1	24.6	0.3	31.9	15.6	20.9
xcomp	0.8	82.9	74.6	78.6	1.2	76.2	73.4	74.8	1.0	40.7	62.9	49.5
ccomp	0.8	72.8	70.8	71.8	0.6	63.1	64.1	63.6	1.2	42.8	40.3	41.5
neg	0.7	89.5	88.1	88.8	0.7	81.2	82.1	81.6	1.1	73.6	72	72.8
iobj	0.7	98.7	91.1	94.7	0.5	96.3	71.0	81.7	1.1	97.1	67.1	79.3
expl	0.6	90.9	84.7	87.7	0.7	87.3	86.8	87.1	0.1	62.5	45	52.3
auxpass	0.5	95.7	96.5	96.1	0.7	98.3	93.5	95.8	1.2	92.3	49.8	64.7
nsubjpass	0.5	94.6	89.9	92.2	0.7	96.1	85.0	90.2	0.6	94.4	67.2	78.5
parataxis	0.4	56.0	32.4	41.1	0.9	52.2	36.8	43.2	0.4	30.4	33.2	31.7
compound	0.4	74.2	66.2	69.9	0.6	72.5	63.6	67.8	4.4	84.7	51.6	64.1
csubj	0.2	77.0	52.5	62.4	0.3	88.1	57.3	69.4	0.2	45.9	31.3	37.2
dep	0.1	70.4	52.4	60.1	0.6	91.2	38.5	54.2	0.5	17.7	16.2	16.9
discourse	0.1	75.6	58.5	66.0	0.1	53.3	60.0	56.5	0.7	77.1	48.4	59.4
foreign	0.0	62.2	69.7	65.7	0.1	98.4	60.7	75.1	0.1	30.9	19.3	23.8
goeswith	0.0	35.7	29.4	32.3	0.1	75.0	19.6	31.1	0.0	26.1	16.7	20.3
csubjpass	0.0	100.0	73.9	85.0	0.0	93.3	71.2	80.8	0.1	87.5	19.7	32.2
list	0.0	–	–	–	0.0	77.0	45.6	57.3	0.1	71.4	18.5	29.4
remnant	0.0	90.0	25.7	40.0	0.0	27.3	10.2	14.8	0.1	92.3	11.8	20.9
reparandum	0.0	–	–	–	0.0	–	–	–	0.1	100.0	34.6	51.4
vocative	0.0	55.6	31.3	40.0	0.0	57.4	52.9	55.1	0.1	84.5	58.6	69.2
dislocated	0.0	88.9	30.8	45.7	0.0	54.5	60.0	57.1	0.0	92.0	48.9	63.9

Table 13: Precision, recall and f-score for different dependency labels for three groups of languages for the universal dependencies experiments in Table 9: G1 (languages with UAS $\geq$ 80), G2 (languages with 70 $\leq$ UAS $<$ 80), G3 (languages with UAS $<$ 70). The rows are sorted by frequency in the G1 languages.

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. 2016a. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016b. One parser, many languages. *arXiv preprint arXiv:1602.01595v1*.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016c. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.

Timothy Baldwin, Jonathan Pool, and Susan M Colowick. 2010. Panlex and LEXTRACT: Translating all words of all languages of the world. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 37–40. Association for Computational Linguistics.

Peter F. Brown, Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Jinho D. Choi, Joel Tetreault, and Amanda Stent. 2015. It depends: Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL*, pages 26–31.

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., July. 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, July.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015a. 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, July. Association for Computational Linguistics.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015b. A neural network model for low-resource universal dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 339–348, Lisbon, Portugal, September. Association for Computational Linguistics.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic 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, July. 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, August. Association for Computational Linguistics.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390, Denver, Colorado, May–June. 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, July. 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.

Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–151, Montréal, Canada, June. Association for Computational Linguistics.

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

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, June. Association for Computational Linguistics.

Percy Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

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, June. Association for Computational Linguistics.

Héctor Martínez Alonso, Željko Agić, Barbara Plank, and Anders Søgaard. 2017. Parsing universal dependencies without training. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 230–240. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 152–159, Stroudsburg, PA, USA. 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., July. Association for Computational Linguistics.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August. Association for Computational Linguistics.

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.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, et al. 2016. Universal Dependencies 1.3. 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.

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, September. Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Joel Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *arXiv preprint arXiv:1503.06733*.

Rudolf Rosa and Zdenek Zabokrtsky. 2015. Klcpos3 - 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, July. Association for Computational Linguistics.

Michael Schlichtkrull and Anders Søgaard. 2017. Cross-lingual dependency parsing with late decoding for truly low-resource languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 220–229. Association for Computational Linguistics.

Karl Stratos, Michael Collins, and Daniel Hsu. 2015. Model-based word embeddings from decompositions of count matrices. 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 1282–1291, Beijing, China, July. 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.

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, June. 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.

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, June. Association for Computational Linguistics.

Jörg Tiedemann. 2015. Improving the cross-lingual projection of syntactic dependencies. In *Nordic Conference of Computational Linguistics NODALIDA 2015*, pages 191–199.

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.

Michael Wick, Pallika Kanani, and Adam Pocock. 2015. Minimally-constrained multilingual embeddings via

artificial code-switching. In *Workshop on Transfer and Multi-Task Learning: Trends and New Perspectives*, Montreal, Canada, December.

Min Xiao and Yuhong Guo. 2015. Annotation projection-based representation learning for cross-lingual dependency parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 73–82, Beijing, China, July. 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, September. Association for Computational Linguistics.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA, June. Association for Computational Linguistics.