Density-Driven Cross-Lingual Transfer of Dependency Parsers

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

Abstract

We present a novel method for the cross-lingual transfer of dependency parsers. Our goal is to induce a dependency parser in a target language of interest without any direct supervision: instead we assume access to parallel translations between the target and one or more source languages, and to supervised parsers in the source language(s). Our key contributions are to show the utility of *dense* projected structures when training the target language parser, and to introduce a novel learning algorithm that makes use of dense structures. Results on several languages show an absolute improvement of 5.51% in average dependency accuracy over the state-of-the-art method of (Ma and Xia, 2014). Our average dependency accuracy of 82.18% compares favourably to the accuracy of fully supervised methods.

1 Introduction

In recent years there has been a great deal of interest in dependency parsing models for natural languages. Supervised learning methods have been shown to produce highly accurate dependency-parsing models; unfortunately, these methods rely on human-annotated data, which is expensive to obtain, leading to a significant barrier to the development of dependency parsers for new languages. Recent work has considered unsupervised methods (e.g. (Klein and Manning, 2004; Headden III et al., 2009; Gillenwater et al., 2011; Mareček and Straka, 2013; Spitkovsky et al., 2013; Le and Zuidema, 2015; Grave and Elhadad, 2015)), or methods that transfer linguistic structures across languages (e.g. (Cohen et al., 2011; McDonald et al., 2011; Ma and Xia, 2014; Tiedemann, 2015; Guo et al., 2015; Zhang and Barzilay, 2015; Xiao and Guo, 2015)), in an effort to reduce or eliminate the need for annotated training examples. Unfortunately the accuracy of these methods generally lags quite substantially behind the performance of fully supervised approaches.

This paper describes novel methods for the transfer of syntactic information between languages. As in previous work (Hwa et al., 2005; Ganchev et al., 2009; McDonald et al., 2011; Ma and Xia, 2014), our goal is to induce a dependency parser in a target language of interest without any direct supervision (i.e., a treebank) in the target language: instead we assume access to parallel translations between the target and one or more source languages, and to supervised parsers in the source languages. We can then use alignments induced using tools such as GIZA++ (Och and Ney, 2000), to transfer dependencies from the source language(s) to the target language (example projections are shown in Figure 1). A target language parser is then trained on the projected dependencies.

Our contributions are as follows:

- We demonstrate the utility of *dense* projected structures when training the target-language parser. In the most extreme case, a “dense” structure is a sentence in the target language where the projected dependencies form a fully projective tree that includes all words in the sentence (we will refer to these structures as “full” trees). In more relaxed definitions, we might include sentences where at least some proportion (e.g., 80%) of the words participate as a modifier in some dependency, or where long sequences (e.g., 7 words or more) of words all participate as modifiers in some dependency. We give empirical evidence that dense structures give particularly high accuracy for their projected dependencies.

[*] Currently on leave at Google Inc. New York.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 328–338,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

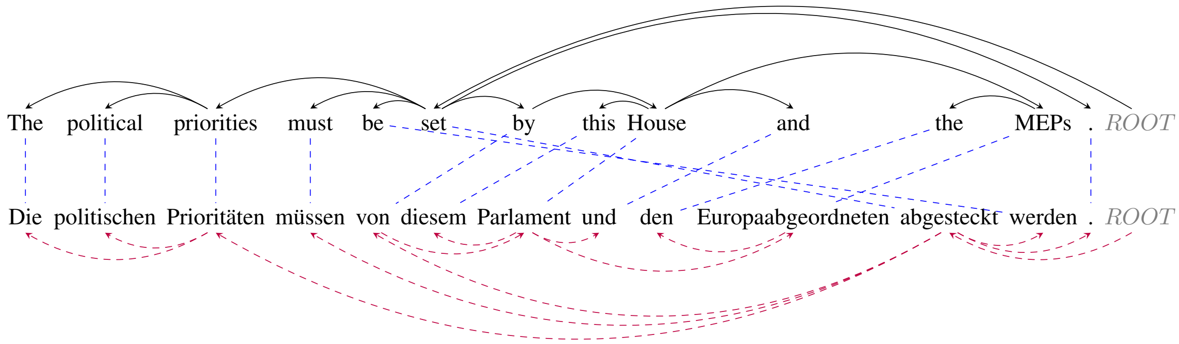

The political priorities must be set by this House and the MEPs . ROOT
Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden . ROOT

Figure 1: An example projection from English to German in the EuroParl data (Koehn, 2005). The English parse tree is the output from a supervised parser, while the German parse tree is projected from the English parse tree using translation alignments from GIZA++.

- We describe a training algorithm that builds on the definitions of dense structures. The algorithm initially trains the model on full trees, then iteratively introduces increasingly relaxed definitions of density. The algorithm makes use of a training method that can leverage partial (incomplete) dependency structures, and also makes use of confidence scores from a perceptron-trained model.

In spite of the simplicity of our approach, our experiments demonstrate significant improvements in accuracy over previous work. In experiments on transfer from a single source language (English) to a single target language (German, French, Spanish, Italian, Portuguese, and Swedish), our average dependency accuracy is 78.89%. When using multiple source languages, average accuracy is improved to 82.18%. This is a 5.51% absolute improvement over the previous best results reported on this data set, 76.67% for the approach of (Ma and Xia, 2014). To give another perspective, our accuracy is close to that of the fully supervised approach of (McDonald et al., 2005), which gives 84.29% accuracy on this data. To the best of our knowledge these are the highest accuracy parsing results for an approach that makes no use of treebank data for the language of interest.

2 Related Work

A number of researchers have considered the problem of projecting linguistic annotations from the source to the target language in a parallel corpus (Yarowsky et al., 2001; Hwa et al., 2005; Ganchev et al., 2009; Spreyer and Kuhn, 2009; McDonald et al., 2011; Ma and Xia, 2014). The projected annotations are then used to train a model in the target language. This prior work involves various innovations such as the use of posterior regularization (Ganchev et al., 2009), the use of entropy regularization and parallel guidance (Ma and Xia, 2014), the use of a simple method to transfer delexicalized parsers across languages (McDonald et al., 2011), and a method for training on partial annotations that are projected from source to target language (Spreyer and Kuhn, 2009). There is also recent work on treebank translation via a machine translation system (Tiedemann et al., 2014; Tiedemann, 2015). The work of (McDonald et al., 2011) and (Ma and Xia, 2014) is most relevant to our own work, for two reasons: first, these papers consider dependency parsing, and as in our work use the latest version of the Google universal treebank for evaluation;[1] second, these papers represent the state of the art in accuracy. The results in (Ma and Xia, 2014) dominate the accuracies for all other papers discussed in this related work section: they report an average accuracy of 76.67% on the languages German, Italian, Spanish, French, Swedish and Portuguese; this evaluation includes all sentence lengths.

Other work on unsupervised parsing has considered various methods that transfer information from source to target languages, where parsers are available in the source languages, but without the use of parallel corpora (Cohen et al., 2011; Dur-

[1] The original paper of (McDonald et al., 2011) does not use the Google universal treebank, however (Ma and Xia, 2014) reimplemented the model and report results on the Google universal treebank.

rett et al., 2012; Naseem et al., 2012; Täckström et al., 2013; Duong et al., 2015; Zhang and Barzilay, 2015). These results are somewhat below the performance of (Ma and Xia, 2014).[2]

3 Our Approach

This section describes our approach, giving definitions of parallel data and of dense projected structures; describing preliminary exploratory experiments on transfer from German to English; describing the iterative training algorithm used in our work; and finally describing a generalization of the method to transfer from multiple languages.

3.1 Parallel Data Definitions

We assume that we have parallel data in two languages. The source language, for which we have a supervised parser, is assumed to be English. The target language, for which our goal is to learn a parser, will be referred to as the "foreign" language. We describe the generalization to more than two languages in §3.5.

We use the following notation. Our parallel data is a set of examples $(e^{(k)}, f^{(k)})$ for $k = 1 \dots n$, where each $e^{(k)}$ is an English sentence, and each $f^{(k)}$ is a foreign sentence. Each $e^{(k)} = e_1^{(k)} \dots e_{s_k}^{(k)}$ where $e_i^{(k)}$ is a word, and s_k is the length of k'th source sentence. Similarly, $f^{(k)} = f_1^{(k)} \dots f_{t_k}^{(k)}$ where $f_j^{(k)}$ is a word, and t_k is the length of k'th foreign sentence.

A **dependency** is a four-tuple (l, k, h, m) where $l \in \{e, f\}$ is the language, k is the sentence number, h is the head index, m is the modifier index. Note that if $l = e$ then we have $0 \leq h \leq s_k$ and $1 \leq m \leq s_k$, conversely if $l = f$ then $0 \leq h \leq t_k$ and $1 \leq m \leq t_k$. We use $h = 0$ when h is the root of the sentence.

For any $k \in \{1 \dots n\}$, $j \in \{0 \dots t_k\}$, $A_{k,j}$ is an integer specifying which word in $e_1^{(k)} \dots e_{s_k}^{(k)}$, word $f_j^{(k)}$ is aligned to. It is NULL if $f_j^{(k)}$ is not aligned to anything. We have $A_{k,0} = 0$ for all k: that is, the root in one language is always aligned to the root in the other language.

In our experiments we use intersected alignments from GIZA++ (Och and Ney, 2000) to provide the $A_{k,j}$ values.

3.2 Projected Dependencies

We now describe various sets of projected dependencies. We use $\mathcal{D}$ to denote the set of all dependencies in the source language: these dependencies are the result of parsing the English side of the translation data using a supervised parser. Each dependency $(l, k, h, m) \in \mathcal{D}$ is a four-tuple as described above, with $l = e$. We will use $\mathcal{P}$ to denote the set of all projected dependencies from the source to target language. The set $\mathcal{P}$ is constructed from $\mathcal{D}$ and the alignment variables $A_{k,j}$ as follows:

$$\begin{aligned}\mathcal{P} = \{(l, k, h, m) : l = f \\ \wedge\ (e, k, A_{k,h}, A_{k,m}) \in \mathcal{D}\}\end{aligned}$$

We say the k'th sentence receives a *full* parse under the dependencies $\mathcal{P}$ if the dependencies (f, k, h, m) for k form a projective tree over the entire sentence: that is, each word has exactly one head, the root symbol is the head of the entire structure, and the resulting structure is a projective tree. We use $\mathcal{T}_{100} \subseteq \{1 \dots n\}$ to denote the set of all sentences that receive a full parse under $\mathcal{P}$. We then define the following set,

$$\mathcal{P}_{100} = \{(l, k, h, m) \in \mathcal{P} : k \in \mathcal{T}_{100}\}$$

We say the k'th sentence receives a *dense* parse under the dependencies $\mathcal{P}$ if the dependencies of the form (f, k, h, m) for k form a projective tree over at least 80% of the words in the sentence. We use $\mathcal{T}_{80} \subseteq \{1 \dots n\}$ to denote the set of all sentences that receive a dense parse under $\mathcal{P}$. We then define the following set,

$$\mathcal{P}_{80} = \{(l, k, h, m) \in \mathcal{P} : k \in \mathcal{T}_{80}\}$$

We say the k'th sentence receives a *span-s* parse where s is an integer if there is a sequence of at least s consecutive words in the target language that are all seen as a modifier in the set $\mathcal{P}$. We use $\mathcal{S}_s$ to refer to the set of all sentences with a span-s parse. We define the sets

$$\mathcal{P}_{\geq 7} = \{(l, k, h, m) \in \mathcal{P} : k \in \mathcal{S}_7\}$$

$$\mathcal{P}_{\geq 5} = \{(l, k, h, m) \in \mathcal{P} : k \in \mathcal{S}_5\}$$

$$\mathcal{P}_{\geq 1} = \{(l, k, h, m) \in \mathcal{P} : k \in \mathcal{S}_1\}$$

Finally, we also create datasets that only include projected dependencies that are consistent with respect to part-of-speech (POS) tags for the head and

[2] With one exception: on Spanish, using the CoNLL definition of dependencies. The good results from (Ma and Xia, 2014) on the universal dependencies for Spanish may show that the result on the CONLL data is an anomaly, perhaps due to the annotation scheme in Spanish being different from other languages.

modifier words in source and target data. We assume a function POS(k, j, i) which returns `TRUE` if the POS tags for words $f_j^{(k)}$ and $e_i^{(k)}$ are consistent. The definition of POS-consistent projected dependencies is then as follows:

$$\bar{\mathcal{P}} = \{(l, k, h, m) \in \mathcal{P} : \text{POS}(k, h, A_{k,h}) \wedge \text{POS}(k, m, A_{k,m})\}$$

We experiment with two definitions for the POS function. The first imposes a hard constraint, that the POS tags in the two languages must be identical. The second imposes a soft constraint, that the two POS tags must fall into the same equivalance class: the equivalence classes used are listed in §4.1.

Given this definition of $\bar{\mathcal{P}}$, we can create sets $\bar{\mathcal{P}}_{100}$, $\bar{\mathcal{P}}_{80}$, $\bar{\mathcal{P}}_{\geq 7}$, $\bar{\mathcal{P}}_{\geq 5}$, and $\bar{\mathcal{P}}_{\geq 1}$, using analogous definitions to those given above.

3.3 Preliminary Experiments with Transfer from English to German

Throughout the experiments in this paper, we used German as the target language for development of our approach. Table 1 shows some preliminary results on transferring dependencies from English to German. We can estimate the accuracy of dependency subsets such as $\mathcal{P}_{100}$, $\mathcal{P}_{80}$, $\mathcal{P}_{\geq 7}$ and so on by comparing these dependencies to the dependencies from a supervised German parser on the same data. That is, we use a supervised parser to provide gold standard annotations. The full set of dependencies $\mathcal{P}$ give 74.0% accuracy under this measure; results for $\mathcal{P}_{100}$ are considerably higher in accuracy, ranging from 83.0% to 90.1% depending on how POS constraints are used.

As a second evaluation method, we can test the accuracy of a model trained on the $\mathcal{P}_{100}$ data. The benefit of the soft-matching POS definition is clear. The hard match definition harms performance, presumably because it reduces the number of sentences used to train the model.

Throughout the rest of this paper, we use the soft POS constraints in all projection algorithms.[3]

3.4 The Training Procedure

We now describe the training procedure used in our experiments. We use a perceptron-trained shift-reduce parser, similar to that of (Zhang and Nivre, 2011). We assume that the parser is able to operate in a "constrained" mode, where it returns the highest scoring parse that is consistent with a given subset of dependencies. This can be achieved via zero-cost dynamic oracles (Goldberg and Nivre, 2013).

```
Inputs: Sets $\mathcal{P}_{100}$, $\mathcal{P}_{80}$, $\mathcal{P}_{\geq 7}$, $\mathcal{P}_{\geq 5}$, $\mathcal{P}_{\geq 1}$ as defined in §3.2.

Definitions: Functions TRAIN, CDECODE, TOP as defined in §3.4.

Algorithm:
  1. $\theta^1 = \text{TRAIN}(\mathcal{P}_{100})$
  2. $\mathcal{P}^1_{100} = \text{CDECODE}(\mathcal{P}_{80} \cup \mathcal{P}_{\geq 7}, \theta^1)$
  3. $\theta^2 = \text{TRAIN}(\mathcal{P}_{100} \cup \text{TOP}(\mathcal{P}^1_{100}, \theta^1))$
  4. $\mathcal{P}^2_{100} = \text{CDECODE}(\mathcal{P}_{80} \cup \mathcal{P}_{\geq 5}, \theta^2)$
  5. $\theta^3 = \text{TRAIN}(\mathcal{P}_{100} \cup \text{TOP}(\mathcal{P}^2_{100}, \theta^2))$
  6. $\mathcal{P}^3_{100} = \text{CDECODE}(\mathcal{P}_{\geq 1}, \theta^3)$
  7. $\theta^4 = \text{TRAIN}(\mathcal{P}_{100} \cup \text{TOP}(\mathcal{P}^3_{100}, \theta^3))$

Output: Parameter vectors $\theta^1, \theta^2, \theta^3, \theta^4$.
```

Figure 2: The learning algorithm.

We assume the following definitions:

- TRAIN$(\mathcal{D})$ is a function that takes a set of dependency structures $\mathcal{D}$ as input, and returns a model θ as its output. The dependency structures are assumed to be full trees: that is, they correspond to fully projected trees with the root symbol as their root.

- CDECODE$(\mathcal{P}, \theta)$ is a function that takes a set of partial dependency structures $\mathcal{P}$, and a model θ as input, and as output returns a set of full trees $\mathcal{D}$. It achieves this by constrained decoding of the sentences in $\mathcal{P}$ under the model θ, where for each sentence we use beam search to search for the highest scoring projective full tree that is consistent with the dependencies in $\mathcal{P}$.

- TOP$(\mathcal{D}, \theta)$ takes as input a set of full trees $\mathcal{D}$, and a model θ. It returns the top m highest scoring trees in $\mathcal{D}$ (in our experiments we used $m = 200,000$), where the score for each tree is the perceptron-based score normalized by the sentence length. Thus we return the

[3] The hard constraint is also used by Ma and Xia (2014).

POS Constraints	$\mathcal{P}$		*dense*		$\mathcal{P}_{100}$		Train on $\mathcal{P}_{100}$
	#sen	Acc.	#sen	Acc.	#sen	Acc.	
No Restriction	$968k$	74.0	$65k$	81.4	$23k$	83.0	69.5
Hard match	$927k$	80.1	$26k$	88.0	$8k$	90.1	68.0
Soft match	$904k$	80.0	$52k$	84.9	$18k$	85.8	70.6

Table 1: Statistics showing the accuracy for various definitions of projected trees: see §3.2 for definitions of $\mathcal{P}$, $\mathcal{P}_{100}$ etc. Columns labeled "Acc." show accuracy when the output of a supervised German parser is used as gold standard data. Columns labeled "#sen" show number of sentences. "dense" shows $\mathcal{P}_{100} \cup \mathcal{P}_{80} \cup \mathcal{P}_{\geq 7}$ and "Train" shows accuracy on test data of a model trained on the $\mathcal{P}_{100}$ trees.

200,000 trees that the perceptron is most confident on.[4]

Figure 2 shows the learning algorithm. It generates a sequence of parsing models, $\theta^1 \ldots \theta^4$. In the first stage of learning, the model is initialized by training on $\mathcal{P}_{100}$. The method then uses this model to fill in the missing dependencies on $\mathcal{P}_{80} \cup \mathcal{P}_{\geq 7}$ using the CDECODE method; this data is added to $\mathcal{P}_{100}$ and the model is retrained. The method is iterated, at each point adding in additional partial structures (note that $\mathcal{P}_{\geq 7} \subseteq \mathcal{P}_{\geq 5} \subseteq \mathcal{P}_{\geq 1}$, hence at each stage we expand the set of training data that is parsed using CDECODE).

3.5 Generalization to Multiple Languages

We now consider the generalization to learning from multiple languages. We again assume that the task is to learn a parser in a single target language, for example German. We assume that we now have multiple source languages. For example, in our experiments with German as the target, we used English, French, Spanish, Portuguese, Swedish, and Italian as source languages. We assume that we have fully supervised parsers for all source languages. We will consider two methods for combining information from the different languages:

Method 1: Concatenation In this approach, we form sets $\mathcal{P}$, $\mathcal{P}_{100}$, $\mathcal{P}_{80}$, $\mathcal{P}_{\geq 7}$ etc. from each of the languages separately, and then concatenate[5] the data to give new definitions of $\mathcal{P}$, $\mathcal{P}_{100}$,$\mathcal{P}_{80}$, $\mathcal{P}_{\geq 7}$ etc.

Method 2: Voting In this case, we assume that each target language sentence is aligned to a source language sentence in each of the source languages. This is the case, for example, in the Europarl data, where we have translations of the same material into multiple languages. We can then create the set $\mathcal{P}$ of projected dependencies using a voting scheme. For any word (k, j) seen in the target language, each source language will identify a headword (this headword may be NULL if there is no alignment giving a dependency). We simply take the most frequent headword chosen by the languages. After creating the set $\mathcal{P}$, we can create subsets such as $\mathcal{P}_{100}$, $\mathcal{P}_{80}$, $\mathcal{P}_{\geq 7}$ in exactly the same way as before.

Once the various projected dependency training sets have been created, we train the dependency parsing model using the algorithm given in §3.4.

4 Experiments

We now describe experiments using our approach. We first describe data and tools used in the experiments, and then describe results.

4.1 Data and Tools

Data We use the EuroParl data (Koehn, 2005) as our parallel data and the Google universal treebank (v2; standard data) (McDonald et al., 2013) as our evaluation data, and as our training data for the supervised source-language parsers. We use seven languages that are present in both Europarl and the Google universal treebank: English (used only as the source language), and German, Spanish, French, Italian, Portuguese and Swedish.

Word Alignments We use Giza++[6] (Och and Ney, 2000) to induce word alignments. Sentences with length greater than 100 and single-word sentences are removed from the parallel data. We follow common practice in training Giza++ for both translation directions, and taking the intersection of the two sets as our final alignment. Giza++ de-

[4] In cases where $|\mathcal{D}| < m$, the entire set $\mathcal{D}$ is returned.

[5] That is, dependency structures projected from different languages are taken to be entirely separate from each other.

[6] `http://www.statmt.org/moses/giza/GIZA++.html`

L	en→trgt				concat→trgt				voting→trgt			
	θ^1	θ^2	θ^3	θ^4	θ^1	θ^2	θ^3	θ^4	θ^1	θ^2	θ^3	θ^4
de	70.56	72.86	73.74	74.32	73.47	75.17	75.59	76.34	78.17	79.29	79.36	79.68
es	75.69	77.27	77.29	78.17	79.53	79.57	79.67	80.28	79.82	80.76	81.16	80.86
fr	77.03	78.54	78.70	79.91	81.23	81.79	82.30	82.24	82.17	82.75	82.47	82.72
it	77.35	78.64	79.06	79.46	81.49	82.25	82.02	82.49	82.58	82.95	83.45	83.67
pt	75.98	77.96	78.29	79.38	80.29	81.73	81.53	82.23	80.12	81.70	81.69	82.07
sv	78.68	80.28	80.81	82.11	82.53	83.78	83.83	83.80	82.85	83.76	83.85	84.06
avg	75.88	77.59	77.98	78.89	79.76	80.72	80.82	81.23	80.95	81.87	82.00	82.18

Table 2: Parsing accuracies of different methods on the test data using the gold standard POS tags. The models $\theta^1 \dots \theta^4$ are described in §3.4. "en→trgt" is the single-source setting with English as the source language. "concat→trgt" and "voting→trgt" are results with multiple source languages for the concatenation and voting methods

fault alignment model is used in all of our experiments.

The Parsing Model For all parsing experiments we use the Yara parser[7] (Rasooli and Tetreault, 2015), a reimplementation of the k-beam arc-eager parser of Zhang and Nivre (2011). We use a beam size of 64, and Brown clustering features[8] (Brown et al., 1992; Liang, 2005). The parser gives performance close to the state of the art: for example on section 23 of the Penn WSJ treebank (Marcus et al., 1993), it achieves 93.32% accuracy, compared to 92.9% accuracy for the parser of (Zhang and Nivre, 2011).

POS Consistency As mentioned in §3.2, we define a soft POS consistency constraint to prune some projected dependencies. A source/target language word pair satisifies this constraint if one of the following conditions hold: 1) the POS tags for the two words are identical; 2) the word forms for the two words are identical (this occurs frequently for numbers, for example); 3) both tags are in one of the following equivalence classes: {ADV ↔ ADJ} {ADV ↔ PRT} {ADJ ↔ PRON} {DET ↔ NUM} {DET ↔ PRON} {DET ↔ NOUN} {PRON ↔ NOUN} {NUM ↔ X} {X ↔ .}. These rules were developed primarily on German, with some additional validation on Spanish. These rules required a small amount of human engineering, but we view this as relatively negligible.

Parameter Tuning We used German as a target language in the development of our approach, and in setting hyper-parameters. The parser is trained using the averaged structured perceptron algorithm (Collins, 2002) with max-violation updates (Huang et al., 2012). The number of iterations over the training data is 5 when training model θ^1 in any setting, and 2, 1 and 4 when training models $\theta^2, \theta^3, \theta^4$ respectively. These values are chosen by observing the performance on German. We use θ^4 as the final output from the training process: this is found to be optimal in English to German projections.

4.2 Results

This section gives results of our approach for the single source, multi-source (concatenation) and multi-source (voting) methods. Following previous work (Ma and Xia, 2014) we use gold-standard part-of-speech (POS) tags on test data. We also provide results with automatic POS tags.

Results with a Single Source Language The first set of results are with a single source language; we use English as the source in all of these experiments. Table 2 shows the accuracy of parameters $\theta^1 \dots \theta^4$ for transfer into German, Spanish, French, Italian, Portuguese, and Swedish. Even the lowest performing model, θ^1, which is trained only on full trees, has a performance of 75.88%, close to the 76.15% accuracy for the method of (Ma and Xia, 2014). There are clear gains as we move from θ^1 to θ^4, on all languages. The average accuracy for θ^4 is 78.89%.

Results with Multiple Source Languages, using Concatenation Table 2 shows results using multiple source languages, using the concatenation method. In these experiments for a given target language we use all other languages in our

[7] `https://github.com/yahoo/YaraParser`

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

Model	$en \rightarrow trgt$	$concat$	$voting$	$sup(1st)$	$sup(ae)$
de	73.01	74.70	78.77	80.29	84.25
es	76.31	78.33	79.17	82.17	84.66
fr	77.54	79.71	80.77	81.33	84.95
it	78.14	80.82	82.03	83.90	87.03
pt	78.14	80.81	80.67	84.80	88.08
sv	79.31	80.81	82.03	81.12	84.87
avg	77.08	79.20	80.57	82.27	85.64

Table 3: Parsing results with automatic part of speech tags on the test data. Sup (1st) is the supervised first-order dependency parser (McDonald et al., 2005) and sup (ae) is the Yara arc-eager parser (Rasooli and Tetreault, 2015).

Model	ge15	zb15	zb_s15	mph11	mx14	$en \rightarrow trgt$	$concat$	$voting$	$sup(1st)$	$sup(ae)$
de	51.0	62.5	74.2	69.77	74.30	$74.32_{(+0.02)}$	$76.34_{(+2.04)}$	$79.68_{(+5.38)}$	81.65	85.34
es	59.2	78.0	78.4	68.72	75.53	$78.17_{(+2.64)}$	$80.28_{(+4.75)}$	$80.86_{(+5.33)}$	83.92	86.69
fr	59.0	78.9	79.6	73.13	76.53	$79.91_{(+3.38)}$	$82.24_{(+5.71)}$	$82.72_{(+6.19)}$	83.51	86.24
it	55.6	79.3	80.9	70.74	77.74	$79.46_{(+1.72)}$	$82.49_{(+4.75)}$	$83.67_{(+5.93)}$	85.47	88.83
pt	57.0	78.6	79.3	69.82	76.65	$79.38_{(+2.73)}$	$82.23_{(+5.58)}$	$82.07_{(+5.42)}$	85.67	89.44
sv	54.8	75.0	78.3	75.87	79.27	$82.11_{(+2.84)}$	$83.80_{(+4.53)}$	$84.06_{(+4.79)}$	85.59	88.06
avg	56.1	75.4	78.4	71.34	76.67	$78.89_{(+2.22)}$	$81.23_{(+4.56)}$	$82.18_{(+5.51)}$	84.29	87.50

Table 4: Comparison to previous work: ge15 (Grave and Elhadad, 2015, Figure 4), zb15 (Zhang and Barzilay, 2015), zb_s15 (Zhang and Barzilay, 2015, semi-supervised with 50 annotated sentences), mph11 (McDonald et al., 2011) and mx14 (Ma and Xia, 2014) on the Google universal treebank v2. The mph11 results are copied from (Ma and Xia, 2014, Table 4). All results are reported on gold part of speech tags. The numbers in parentheses are absolute improvements over (Ma and Xia, 2014). Sup (1st) is the supervised first-order dependency parser used by (Ma and Xia, 2014) and sup(ae) is the Yara arc-eager supervised parser (Rasooli and Tetreault, 2015).

data as source languages. The performance of θ^1 improves from an average of 75.88% for a single source language, to 79.76% for multiple languages. The performance of θ^4 gives an additional improvement to 81.23%.

Results with Multiple Source Languages, using Voting The final set of results in Table 2 are for multiple languages using the voting strategy. There are further improvements: model θ^1 has average accuracy of 80.95%, and model θ^4 has average accuracy of 82.18%.

Results with Automatic POS Tags We use our final θ^4 models to parse the treebank with automatic tags provided by the same POS tagger used for tagging the parallel data. Table 3 shows the results for the transfer methods and the supervised parsing models of (McDonald et al., 2011) and (Rasooli and Tetreault, 2015). The first-order supervised method of (McDonald et al., 2005) gives only a 1.7% average absolute improvement in accuracy over the voting method. For one language (Swedish), our method actually gives improved accuracy over the 1st order parser.

Comparison to Previous Results Table 4 gives a comparison of the accuracy on the six languages, using the single source and multiple source methods, to previous work. As shown in the table, our model outperforms all models: among them, the results of (McDonald et al., 2011) and (Ma and Xia, 2014) are directly comparable to us because they use the same training and evaluation data. The recent work of (Xiao and Guo, 2015) uses the same parallel data but evaluates on CoNLL treebanks but their results are lower than Ma and Xia (2014). The recent work of (Guo et al., 2015) evaluates on the same data as ours but uses different parallel corpora. They only reported on three languages (German: 60.35, Spanish: 71.90 and French: 72.93) which are all far bellow our results. The work of (Grave and Elhadad, 2015) is the state-of-the-art fully unsupervised model with

L	$en \rightarrow trg$							concat							voting						
	$\mathcal{P}_{80} \cup \mathcal{P}_{\geq 7}$				$\mathcal{P}_{100}$			$\mathcal{P}_{80} \cup \mathcal{P}_{\geq 7}$				$\mathcal{P}_{100}$			$\mathcal{P}_{80} \cup \mathcal{P}_{\geq 7}$				$\mathcal{P}_{100}$		
	sen#	dep#	len	acc.	sen#	len	acc.	sen#	dep#	len	acc.	sen#	len	acc.	sen#	dep#	len	acc.	sen#	dep#	acc.
de	34k	9.6	28.3	84.7	18k	6.8	85.8	98k	9.4	28.8	84.1	51k	6.3	88.0	75k	10.8	23.5	84.5	47k	8.2	91.4
es	108k	10.9	31.4	87.3	20k	7.4	89.4	536k	11.0	31.8	86.3	89k	7.5	89.8	346k	17.0	28.5	86.1	109k	12.1	89.2
fr	70k	10.1	32.8	85.8	13k	6.7	84.1	342k	10.5	33.0	87.5	47k	6.9	89.5	303k	14.9	29.9	87.4	78k	11.7	91.2
it	57k	10.0	31.2	84.4	9k	6.3	76.9	434k	11.1	31.3	84.7	70k	7.4	87.2	301k	15.2	28.5	84.5	101k	12.4	87.9
pt	489k	10.0	31.0	85.2	10k	6.0	84.0	462k	11.1	31.3	81.4	77k	7.3	85.4	222k	12.4	30.3	81.3	39k	8.8	85.8
sv	81k	10.4	25.8	83.1	30k	7.4	87.8	255k	9.5	23.6	84.6	79k	6.8	89.7	211k	12.2	25.2	84.2	86k	9.5	88.8
avg	140k	10.2	30.1	85.1	17k	6.8	84.7	354k	10.4	30.0	84.8	69k	7.0	88.3	243k	13.7	27.6	84.7	77k	10.4	89.0

Table 5: Table showing statistics on projected dependencies for the target languages, for the single-source, multi-source (concat) and multi-source (voting) methods. "sen#" is the number of sentences. "dep#" is the average number of dependencies per sentence. "len" is the average sentence length. "acc." is the percentage of projected dependencies that agree with the output from a supervised parser.

minimal linguistic prior knowledge. The model of (Zhang and Barzilay, 2015) does not use any parallel data but uses linguistic information across languages. Their semi-supervised model selectively samples 50 annotated sentences but our model outperforms their model.

Compared to the results of (McDonald et al., 2011) and (Ma and Xia, 2014) which are directly comparable, there are clear improvements across all languages; the highest accuracy, 82.18%, is a 5.51% absolute improvement over the average accuracy for (Ma and Xia, 2014).

5 Analysis

We conclude with some analysis of the accuracy of the projected dependencies for the different languages, for different definitions ($\mathcal{P}_{100}$, $\mathcal{P}_{80}$ etc.), and for different projection methods. Table 5 gives a summary of statistics for the various languages. Recall that German is used as the development language in our experiments; the other languages can be considered to be test languages. In all cases the accuracy reported is the percentage match to a supervised parser used to parse the same data.

There are some clear trends. The accuracy of the $\mathcal{P}_{100}$ datasets is high, with an average accuracy of 84.7% for the single source method, 88.3% for the concatenation method, and 89.0% for the voting method. The voting method not only increases accuracy over the single source method, but also increases the number of sentences (from an average 17k to 77k) and the average number of dependencies per sentence (from 6.8 to 10.4).

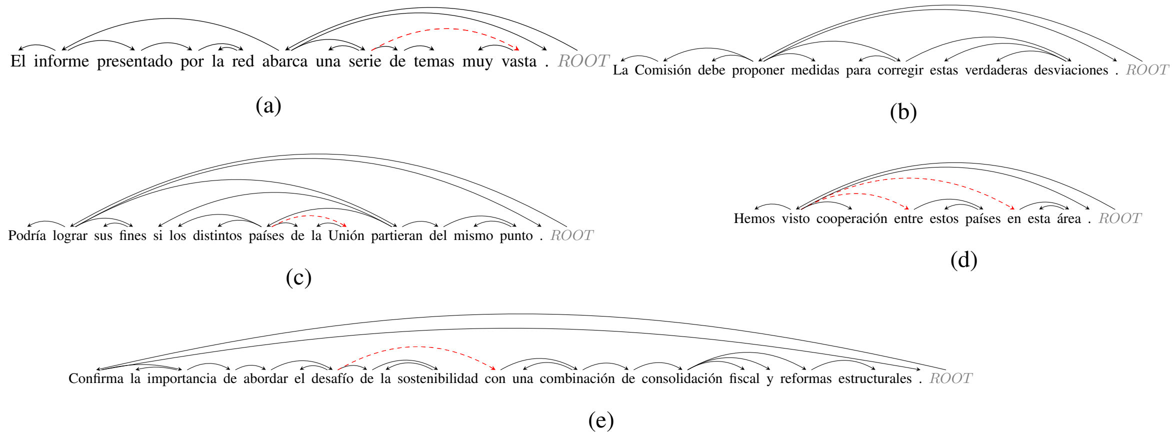
The accuracy of the $\mathcal{P}_{80} \cup \mathcal{P}_{\geq 7}$ datasets is slightly lower, with around 83-87% accuracy for the single source, concatenation and voting methods. The voting method gives a significant increase in the number of sentences—from an average of 140k to 243k. The average sentence length for this data is around 28 words, considerably longer than the $\mathcal{P}_{100}$ data; the addition of longer sentences is very likely beneficial to the model. For the voting method the average number of dependencies is 13.7, giving an average density of 50% on these sentences.

The accuracy for the different languages, in particular for the voting data, is surprisingly uniform, with a range of 85.8-91.4% for the $\mathcal{P}_{100}$ data, and 81.3-87.4% for the $\mathcal{P}_{80} \cup \mathcal{P}_{\geq 7}$ data. The number of sentences for each language, the average length of those sentences, and average number of dependencies per sentence is also quite uniform, with the exception of German, which is a clear outlier. German has fewer sentences, and fewer dependencies per sentence: this may account for it having the lowest accuracy for our models. Future work should investigate why this is the case: one hypothesis is that German has quite different word order from the other languages (it is V2, and verb final), which may lead to a degradation in the quality of the alignments from GIZA++, or in the projection process.

Finally, figure 3 shows some randomly selected examples from the $\mathcal{P}_{100}$ data for Spanish, giving a qualitative feel for the data obtained using the voting method.

6 Conclusions

We have described a density-driven method for the induction of dependency parsers using parallel data and source-language parsers. The key ideas are a series of increasingly relaxed definitions of density, together with an iterative training procedure that makes use of these definitions. The method gives a significant gain over previous methods, with dependency accuracies approach-


El informe presentado por la red abarca una serie de temas muy vasta . ROOT
La Comisión debe proponer medidas para corregir estas verdaderas desviaciones . ROOT
Podría lograr sus fines si los distintos países de la Unión partieran del mismo punto . ROOT
Hemos visto cooperación entre estos países en esta área . ROOT
Confirma la importancia de abordar el desafío de la sostenibilidad con una combinación de consolidación fiscal y reformas estructurales . ROOT

(a)

(b)

(c)

(d)

(e)

Figure 3: Randomly selected examples of Spanish dependency structures derived using the *voting* method. Dashed/red dependencies are mismatches with the output of a supervised Spanish parser; all other dependencies match the supervised parser. In these examples, 92.4% of dependencies match the supervised parser; this is close to the average match rate on Spanish of 89.2% for the voting method.

ing the level of fully supervised methods. Future work should consider application of the method to a broader set of languages, and application of the method to transfer of information other than dependency structures.

Acknowledgement

We thank Avner May and anonymous reviewers for their useful comments. Mohammad Sadegh Rasooli was supported by a grant from Bloomberg's Knowledge Engineering team.

References

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.

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.

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, July. 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.

Jennifer Gillenwater, Kuzman Ganchev, João Graça, Fernando Pereira, and Ben Taskar. 2011. Posterior sparsity in unsupervised dependency parsing. *The Journal of Machine Learning Research*, 12:455–490.

Yoav Goldberg and Joakim Nivre. 2013. Training deterministic parsers with non-deterministic oracles. *TACL*, 1:403–414.

Edouard Grave and Noémie Elhadad. 2015. A convex and feature-rich discriminative approach to dependency grammar induction. 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 1375–1384, Beijing, China, July. 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.

William P. Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 101–109, Boulder, Colorado, June. Association for Computational Linguistics.

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.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, 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.

Phong Le and Willem Zuidema. 2015. Unsupervised dependency parsing: Let's use supervised parsers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 651–661, Denver, Colorado, May–June. Association for Computational Linguistics.

Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. 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.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational linguistics*, 19(2):313–330.

David Mareček and Milan Straka. 2013. Stop-probability estimates computed on a large corpus improve unsupervised dependency parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 281–290, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523–530, 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, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of 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.

Franz Josef Och and Hermann Ney. 2000. Giza++: Training of statistical translation models.

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

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2013. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1983–1995, Seattle, Washington, USA, October. Association for Computational Linguistics.

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

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. *Transactions for ACL.*

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.

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.

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.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, September.

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.