Journal of Artificial Intelligence Research 55 (2016) 995-1023 Submitted 08/15; published 04/16

A Distributed Representation-Based Framework for Cross-Lingual Transfer Parsing

Jiang Guo JGUO@IR.HIT.EDU.CN
Wanxiang Che CAR@IR.HIT.EDU.CN
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology
Harbin, Heilongjiang, China

David Yarowsky YAROWSKY@JHU.EDU
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD, USA

Haifeng Wang WANGHAIFENG@BAIDU.COM
Baidu Inc., Beijing, China

Ting Liu TLIU@IR.HIT.EDU.CN
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology
Harbin, Heilongjiang, China

Abstract

This paper investigates the problem of cross-lingual transfer parsing, aiming at inducing dependency parsers for low-resource languages while using only training data from a resource-rich language (e.g., English). Existing model transfer approaches typically don't include lexical features, which are not transferable across languages. In this paper, we bridge the *lexical feature gap* by using distributed feature representations and their composition. We provide two algorithms for inducing cross-lingual distributed representations of words, which map vocabularies from two different languages into a common vector space. Consequently, both lexical features and non-lexical features can be used in our model for cross-lingual transfer. Furthermore, our framework is flexible enough to incorporate additional useful features such as cross-lingual word clusters. Our combined contributions achieve an average relative error reduction of 10.9% in labeled attachment score as compared with the delexicalized parser, trained on English universal treebank and transferred to three other languages. It also significantly outperforms state-of-the-art delexicalized models augmented with projected cluster features on identical data. Finally, we demonstrate that our models can be further boosted with minimal supervision (e.g., 100 annotated sentences) from target languages, which is of great significance for practical usage.

1. Introduction

Dependency Parsing has been one of the long-standing central problems in natural language processing (NLP). The goal of dependency parsing is to induce implicit tree structures for natural language sentence following the dependency grammar, which can be highly beneficial for various downstream tasks, such as question answering, machine translation and knowledge mining/representation. The majority of work on dependency parsing has been dedicated to resource-rich languages, such as English and Chinese. For these languages, there exists large-scale annotated treebanks that can be used for supervised training of dependency parsers, such as the Penn Treebank (Marcus, Marcinkiewicz, & Santorini, 1993; Xue, Xia, Chiou, & Palmer, 2005). However, for most of the languages in the world, there are very few or even no labeled training data for parsing, and it is both labor intensive and time consuming to manually annotate treebanks for all languages. This fact has given rise to a range of research on unsupervised methods (Klein & Manning, 2004), and transfer methods (Hwa, Resnik, Weinberg, Cabezas, & Kolak, 2005; McDonald, Petrov, & Hall, 2011) for linguistic structure prediction.

Considering that the unsupervised methods fall far behind the transfer methods in terms of accuracy, as well as the difficulty in evaluation, we will focus on the transfer methods in this study. We attempt to build parsers for low-resource languages by exploiting treebanks from resource-rich languages. There are two approaches to linguistic transfer in general, namely *data transfer* and *model transfer*. Data transfer methods emphasizes the creation of artificial training data that can be used for supervised training on the target language side. They have the appealing property that they can learn language-specific linguistic structures effectively. The major drawbacks are the requirement of parallel data and the noise in the automatically created training data introduced by word alignment-based projection. On the other hand, model transfer methods build models on the source language side, which are used directly for parsing target languages without the need of creating annotated data in target languages.

This paper falls into the latter category. The major obstacle in transferring a parsing system from one language to another is the lexical features (e.g., words) that are not directly transferable across languages. To address this challenge, McDonald et al. (2011) built a *delexicalized* parser – a parser that only has non-lexical features. A delexicalized parser makes sense in that POS tag features are significantly predictive for unlabeled dependency parsing. However, for labeled dependency parsing, especially for semantic-oriented dependencies like Stanford typed dependencies (De Marneffe et al., 2006; De Marneffe & Manning, 2008), these non-lexical features are not predictive enough. Täckström, McDonald, and Uszkoreit (2012) proposed to learn cross-lingual word clusters from multilingual paralleled unlabeled data through word alignments, and apply these clusters as features for semi-supervised delexicalized parsing. Word clusters can be thought of as a kind of coarse-grained representations of words. Thus, this approach partially fills the gap of lexical features in cross-lingual learning of dependency parsing.

This paper proposes a novel approach for cross-lingual dependency parsing that is based on pure distributed feature representations. In contrast to the discrete feature representations used in traditional dependency parsers, distributed representations map symbolic features into a continuous representation space, that can be shared across languages. Therefore, our model has the ability to utilize both lexical and non-lexical features naturally. Specifically, our framework contains two primary components:

- A neural network-based dependency parser. We expect a non-linear model for dependency parsing in our study, because distributed feature representations are shown to be more effective in non-linear architectures than in linear architectures (Wang & Manning, 2013). Chen and Manning (2014) proposed a transition-based dependency parser using a neural network architecture, which is simple but works well on benchmark datasets. Briefly, this model simply replaces the predictor in transition-based dependency parser with a well-designed neural network classifier. We will provide explanations for the merits of this model in Section 3, as well as how we adapt it to the cross-lingual task.

- Cross-lingual word representation learning. The key to filling the *lexical feature gap* is to project the representations of these features from different languages into a common vector space, preserving the translational equivalence. We will study and compare two approaches of learning cross-lingual word representations in Section 4. The first approach is named *robust projection*, and the second approach is based on *canonical correlation analysis*. Both approaches are simple to implement and are scalable to large data.

Another drawback of the model transfer methods is that they focus only on the universal structures across various languages, and thus lack the ability of recovering the target language-specific structures. Therefore, it is necessary to conduct target language adaptation on the top of the transferred models. We introduce a practical and straightforward solution by incorporating minimal supervision from target languages (Section 6).

We evaluate our models on the universal multilingual treebanks v2.0 (McDonald et al., 2013). Case studies include transferring from English (EN) to German (DE), Spanish (ES) and French (FR). Experiments show that by incorporating lexical features, the performance of cross-lingual dependency parsing can be improved significantly. By further embedding cross-lingual cluster features (Täckström et al., 2012), we achieve an average relative error reduction of 10.9% in labeled attachment score (LAS), as compared with the delexicalized parsers. It also significantly outperforms the delexicalized models of McDonald et al. augmented with cluster features on identical data. In addition, we show that by using a small amount of labeled training data (e.g., 100 sentences) at the target language side for parameter adaptation (minimal supervision), the performance of our cross-lingual transfer system can be boosted, and the recalls of language-specific dependency structures are improved dramatically.[1]

The original major contributions of this paper include:

- We propose a novel and flexible cross-lingual learning framework for dependency parsing based on distributed representations, which can effectively incorporate both lexical and non-lexical features.

- We present two novel and effective approaches for inducing cross-lingual word representation that bridge the *lexical feature gap* in cross-lingual dependency parsing transfer.

- We show that cross-lingual word cluster features can be effectively embedded into our model, leading to significant additive improvements.

- We show that the our cross-lingual transfer systems can be easily and effectively adapted to target languages with minimal supervision, demonstrating great potential in practical usage.

2. Background

This section describes the necessary background which is crucial for understanding our transfer parsing framework.

[1] This article is a thoroughly revised and extended version of the work of Guo, Che, Yarowsky, Wang, and Liu (2015). We provide a more detailed linguistic and methodological background of cross-lingual parsing. Additional extensions primarily include experiments and analysis of target language adaptation with minimal supervision. Our system is made publicly available at: `https://github.com/jiangfeng1124/acl15-clnndep`.

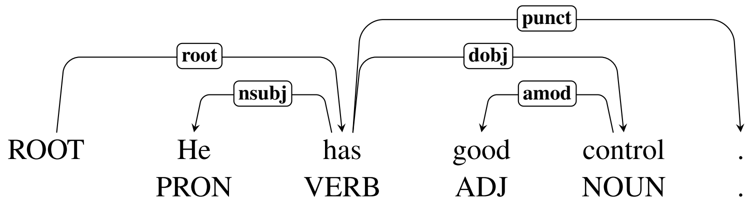

Figure 1: An example labeled dependency tree.

2.1 Dependency Parsing

Given an input sentence $\mathbf{x} = w_0 w_1 ... w_n$ where w_i is the i^{th} word of $\mathbf{x}$, the goal of dependency parsing is to build a dependency tree, which can be denoted by $\mathbf{d} = \{(h, m, l) : 0 \le h \le n; 0 < m \le n, l \in \mathcal{L}\}$, where (h, m, l) indicates a directed arc from the head word w_h to its modifier w_m with a dependency label l, and $\mathcal{L}$ is the label set (Figure 1).

The mainstream models that have been proposed for dependency parsing can be described as either graph-based models or transition-based models (McDonald & Nivre, 2007). Graph-based models (Eisner, 1996; McDonald, Crammer, & Pereira, 2005) view the parsing problem as finding the highest scoring tree from a directed graph. The score of a dependency tree is typically factored into scores of some small independent structures. The way of factorization defines the *order* of a model and also the complexity in the inference process (McDonald & Pereira, 2006; Carreras, 2007; Koo & Collins, 2010). For instance, *first-order* models are factored into dependency arcs, thus also known as *arc-factored* models. *Higher-order* models would consider more expressive substructures such as sibling and grandchild structures. Transition-based models instead aim to predict a transition sequence from an initial parser state to some terminal states, conditioned on the parsing history (Yamada & Matsumoto, 2003; Nivre, 2003; Nivre, Hall, & Nilsson, 2004). This approach has a lot of interest since it is fast (linear time for projective parsing) and can incorporate rich non-local features (Zhang & Nivre, 2011).

It has been considered in the past that simple transition-based parsing using greedy decoding and local training is not as accurate as graph-based parsers that are globally trained and use exact inference algorithms. However, Chen and Manning (2014) showed that the greedy transition-based parsers can be significantly improved with a well-designed neural network architecture. This approach can be considered as a new paradigm of parsing, in that it is based on pure distributed feature representations. More recently, this architecture has been improved in different ways. For example, Weiss, Alberti, Collins, and Petrov (2015) combined the neural network with structured perceptron, and use beam-search for decoding, achieving the new state-of-the-art performance. Dyer, Ballesteros, Ling, Matthews, and Smith (2015) instead explored novel techniques for learning better representations of parser states by utilizing long short-term memory networks (LSTM). Other work also includes that of Zhou, Zhang, Huang, and Chen (2015) who applied structured learning with beam-search decoding over the neural network model. In this study, we choose the original Chen & Manning's architecture, without losing generality, to build our basic dependency parsing models for cross-lingual transfer.

2.2 Distributed Representations for NLP

Recent years have seen numerous attempts of learning distributed representations for different natural language objects, from morphemes, words and phrases, to sentences and documents. Using distributed representations, these symbolic units are embedded into a dense, continuous and low-dimensional vector space, thus it is often referred to as *embeddings*.[2]

Distributed representation is attractive in NLP for several reasons. First, it provides a straightforward way of measuring the similarities between natural language objects. Through distributed representations, we can easily tell which two words/phrases/documents are similar in semantic or even other aspects by simply measuring the *cosine* distance of vectors.

Second, it can be learned from large-scale unannotated data in general, and thus can be highly beneficial for various downstream applications as a source to alleviate data sparsity. The most straightforward way of applying distributed representations to NLP tasks is to fed the distributed feature representations into existing supervised NLP systems as augmented features, in a semi-supervised fashion (Turian, Ratinov, & Bengio, 2010). Despite the simplicity and effectiveness, it has been shown that the potential of distributed representations cannot be fully exploited in the generalized linear models which are adopted in most of the traditional NLP systems (Wang & Manning, 2013). One remedy is to discretize the distributed feature representations, that is to convert the continuous, dense and low-dimensional vectors into traditional discrete, sparse and high-dimensional space, as studied by Guo, Che, Wang, and Liu (2014). However, we believe that a non-linear system (e.g., neural network) is a more powerful and promising solution. Some decent progress has already been made in this paradigm of NLP on various tasks, such as neural sequence labeling (Collobert et al., 2011), dependency parsing (Chen & Manning, 2014), sentence classification (Kim, 2014) and machine translation (Sutskever, Vinyals, & Le, 2014).

Third, it provides such a kind of representation that can be shared across languages, tasks and even diverse modalities of data resources. This property has motivated lines of research on multilingual representation learning (Klementiev et al., 2012; Chandar A P et al., 2014; Hermann & Blunsom, 2014), multi-task learning (Collobert & Weston, 2008) and multi-modal learning (Srivastava & Salakhutdinov, 2012). This is also the primary motivation of this work that facilitates cross-lingual transfer parsing via multilingual distributed representation learning of words.

3. Cross-Lingual Dependency Parsing

In this section, we first describe the primary transition-based dependency parsing model utilizing neural networks, and then details for cross-lingual transfer.

3.1 A Neural Network Architecture for Transition-Based Dependency Parsing

In this section, we first briefly describe transition-based dependency parsing and the *arc-standard* parsing algorithm. Then we revisit the neural network architecture for transition-based dependency parsing proposed by Chen and Manning (2014).

As discussed in Section 2.1, transition-based parsing generates a dependency tree by predicting a transition sequence from an initial parser state to the terminal state. Several transition-based parsing algorithms have been presented in the literature, such as the *arc-standard* and *arc-eager* algorithms for projective parsing (Nivre, 2003, 2004), the *list-based* algorithm (Nivre, 2008) and the

[2] In this paper, these two terminologies are used interchangeably.

swap-based algorithm (Nivre, 2009) for non-projective parsing. Different algorithms have different transition actions. Take the *arc-standard* algorithm for example, each parsing state (typically known as *configuration*) can be represented as a tuple consisting of a *stack* S, a *buffer* B, and a partially derived forest (i.e., a set of dependency arcs) A. Given an input word sequence $\mathbf{x} = w_1 w_2, ..., w_n$, the initial *configuration* can be represented as: $\langle [w_0]_S, [w_1 w_2, ..., w_n]_B, \varnothing \rangle$, and the terminal *configuration* is $\langle [w_0]_S, []_B, A \rangle$, where w_0 is a pseudo word indicating the *root* of the whole dependency tree. Denoting S_i ($i = 0, 1, ...$) as the i^{th} element in the *stack*, and B_i ($i = 0, 1, ...$) as the i^{th} element in the *buffer*,[3] the *arc-standard* system defines three types of transition actions: LEFT-ARC(r), RIGHT-ARC(r), and SHIFT, r is the dependency relation.

- LEFT-ARC(r): extend A with a new arc ($S_1 \xleftarrow{r} S_0$) (S_0 the head and S_1 the modifier) and remove S_1 from the *stack*.
- RIGHT-ARC(r): extend A with a new arc ($S_1 \xrightarrow{r} S_0$) (S_1 the head and S_0 the modifier) and pop S_0 from the *stack*.
- SHIFT: move B_0 from the *buffer* to the *stack*. Precondition is that B is not empty.

The typical approach for greedy *arc-standard* parsing is to build a multi-class classifier (e.g., support vector machines, maximum entropy models) of predicting the transition action given a feature vector extracted from a specific *configuration*. While conventional feature engineering suffers from the problem of *sparsity*, *incompleteness* and *expensive feature computation* (Chen & Manning, 2014), the neural network model provides an effective solution.

The architecture of the neural network based dependency parsing model is illustrated in Figure 2. Unlike the high-dimensional, sparse and discrete features used by traditional parsing models, in the neural network model, we apply distributed feature representations. Primarily, three types of information are extracted from a *configuration* in Chen & Manning's model: word features, POS features and relation features respectively. In this study, we add non-local features including *distance* features indicating the distance between two items, and the *valency* features indicating the number of children for a given item (Zhang & Nivre, 2011). Both *distance* and *valency* features are discretized into buckets. All of these features are then projected to an embedding layer via corresponding *lookup tables* (i.e., embedding matrices), which will be estimated through the training process. The complete feature templates used in our system are shown in Table 1.

Then, feature compositions are performed at the hidden layer via the *cube activation function*:

$$\mathbf{h} = g(\mathbf{x}) = (\mathbf{W}_1 \cdot [x^w, x^t, x^r, x^d, x^v] + \mathbf{b}_1)^3$$

where $\mathbf{W}_1$ is the weight matrix from the input layer to the hidden layer, and $\mathbf{b}_1$ is the bias vector.

Feature compositions are important not only in dependency parsing but in NLP in general. Researchers used to do cost-intensive manual feature engineering to design a large set of feature templates. However, this approach cannot cover all potentially useful features. Lei, Xin, Zhang, Barzilay, and Jaakkola (2014) showed that a full feature representation can be derived from the *Kronecker product* of multiple views of features, which results in a tensor model. By representing the tensor in a low-rank form using CANDECOMP/PARAFAC (CP) tensor decomposition (Kolda & Bader, 2009), the number of parameters can be effectively reduced, and thus is suitable for tasks with limited training data (Cao & Khudanpur, 2014).

[3] S_0/B_0 is the top/head element of the stack/buffer.

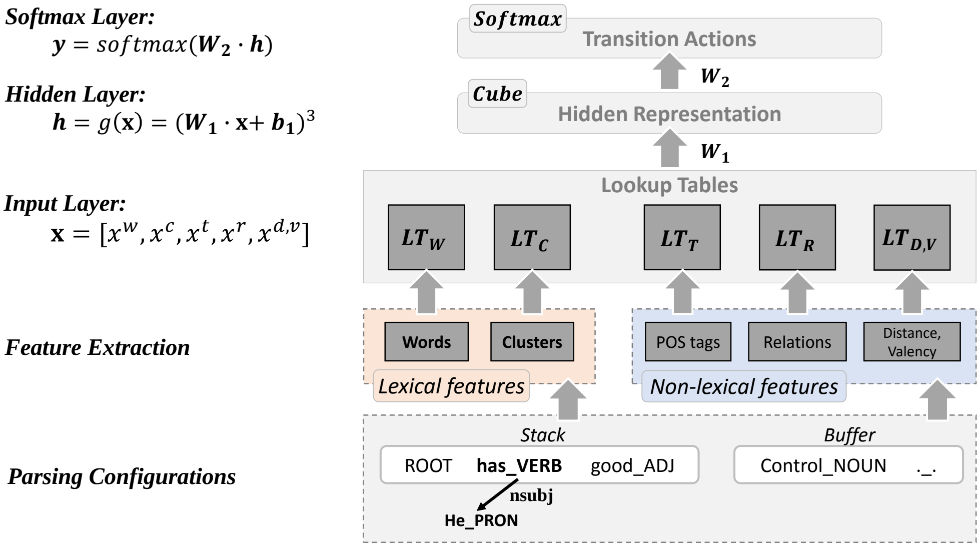

Figure 2: Neural network model for dependency parsing. The *Cluster* features are introduced in Section 5.2 and 5.3.

Type	**Feature Templates**
Word[†]	$E^w_{S_i}, E^w_{B_i}, i = 0, 1, 2$ $E^w_{lc1(S_i)}, E^w_{rc1(S_i)}, E^w_{lc2(S_i)}, E^w_{rc2(S_i)}, i = 0, 1$ $E^w_{lc1(lc1(S_i))}, E^w_{rc1(rc1(S_i))}, i = 0, 1$
POS[‡]	$E^t_{S_i}, E^t_{B_i}, i = 0, 1, 2$ $E^t_{lc1(S_i)}, E^t_{rc1(S_i)}, E^t_{lc2(S_i)}, E^t_{rc2(S_i)}, i = 0, 1$ $E^t_{lc1(lc1(S_i))}, E^t_{rc1(rc1(S_i))}, i = 0, 1$
Relation[‡]	$E^r_{lc1(S_i)}, E^r_{rc1(S_i)}, E^r_{lc2(S_i)}, E^r_{rc2(S_i)}, i = 0, 1$ $E^r_{lc1(lc1(S_i))}, E^r_{rc1(rc1(S_i))}, i = 0, 1$
Distance[‡]	$E^d_{\langle S_0, S_1 \rangle}, E^d_{\langle S_0, B_0 \rangle}$
Valency[‡]	$E^{lv}_{S_0}, E^{lv}_{S_1}, E^{rv}_{S_1}$

Table 1: Feature templates of the neural network model for transition-based dependency parsing. $E_p^{\{w,c,t,r,d,lv,rv\}}$ indicates various feature embeddings of the element at position p. $lc1$ ($rc1$) is the first child to the left (right) and $lc2$ ($rc2$) is the second child to the left (right). [†] indicates the lexical features, [‡] indicates the non-lexical features.

We suggest that the cube activation function $g(x) = x^3$ can be viewed as a special case of the low-rank tensor. For verification, $g(x)$ can be expanded as:

$$g(w_1 x_1 + \ldots + w_m x_m + b) = \\ \sum_{i,j,k} (w_i w_j w_k) x_i x_j x_k + \sum_{i,j} b(w_i w_j) x_i x_j + \ldots$$

If we treat the bias term as $b \times x_0$ where $x_0 = 1$, then the weight corresponding to each feature combination $x_i x_j x_k$ can be wrote as $w_i w_j w_k$, which is exactly the same as a rank-1 component tensor in the low-rank form using CP tensor decomposition. Consequently, the cube activation function implicitly derives full feature combinations. In fact, we can add as many features as possible to the input layer to improve the parsing accuracy. We will show in Section 5.2 that the Brown-cluster features can be readily incorporated into our model.

The composed features are then propagated to the output layer, generating a probabilistic distribution of the output labels (i.e., transition actions) via the `softmax` activation function: $\mathbf{y} = softmax(\mathbf{W}_2 \cdot \mathbf{h})$. We use the following objective function to train the model:

$$\mathcal{J}(\theta) = \frac{1}{N} \sum_{i=0}^{N} CrossEnt(\mathbf{d}_i, \mathbf{y}_i) + \frac{\lambda}{2} \|\theta\|^2$$

where $CrossEnt(\mathbf{p}, \mathbf{q})$ is the cross-entropy between two distributions $\mathbf{p}$ and $\mathbf{q}$:

$$CrossEnt(\mathbf{p}, \mathbf{q}) = \sum_k \mathbf{p}_k \ln \mathbf{q}_k$$

All parameters in θ are trained using back-propagation. In this model, θ typically consists of all the embedding matrices and weights in the network. However, in some cases, θ may exclude the word embedding matrix E^w, which indicates that the word embeddings are constrained to be **fixed** (i.e., without updating) while training.

3.2 Cross-Lingual Transfer

The idea of cross-lingual transfer using the parser we examined above is straightforward. In contrast to traditional approaches that have to discard rich lexical features (delexicalizing) when transferring models from one language to another, our model can be transferred using the full model trained on the source language side (i.e., English).

Since the non-lexical feature (POS, relation, distance, valency) embeddings are directly transferable between languages, the key component of this framework is the cross-lingual learning of lexical feature embeddings (i.e., word embeddings). Once the cross-lingual word embeddings are induced, we first learn a dependency parser at the source language side. After that, the parser will be directly used for parsing target language data.

3.2.1 UNIVERSAL DEPENDENCIES

As discussed previously, cross-lingual model transfer assumes universal grammatical structures that can be identified in multiple languages. Therefore, when evaluated on the test set of target language with either unlabeled attachment score (UAS) or labeled attachment score (LAS), the performance of transfer parsing rely heavily on the multilingual consistency of annotation schemes. Generally syntactic annotation schemes differ in the *head-finding rules* (e.g., the choice of lexical versus functional head) and the *dependency relation labels* (i.e., the syntactic tagset). It is a challenging task to construct multilingual treebanks with such consistent annotations. In the initial cross-lingual parsing studies, the CoNLL shared task datasets (Buchholz & Marsi, 2006) are broadly used. However, inconsistencies occur both in the head-finding rules and the syntactic tagset across languages, which made it difficult to evaluate the cross-lingual parsers.

In order to overcome these difficulties, a new collection of multilingual treebanks with homogeneous syntactic dependency annotation has been presented recently, namely the Universal Dependency Treebanks (UDT) (McDonald et al., 2013). The universal annotation scheme was created by harmonizing available treebanks in slightly different variants of the Stanford typed dependencies (De Marneffe et al., 2006), along with the universal Part-of-Speech tags (Petrov, Das, & McDonald, 2012). This dataset greatly facilitates research on multilingual syntactic analysis, and also makes it possible to use LAS for evaluation. In fact, UDT has already been used as a standard dataset for benchmarking research on cross-lingual transfer parsing (Ma & Xia, 2014; Tiedemann, 2014; Zhang & Barzilay, 2015; Duong, Cohn, Bird, & Cook, 2015a, 2015b; Rasooli & Collins, 2015). Other efforts towards universal dependencies include the most recent Universal Dependencies project (UD) [4] and HamleDT (Zeman et al., 2014). In this paper, we conduct experiments on the UDT (v2.0) [5] dataset without losing generality.

3.2.2 PROJECTIVE VS. NON-PROJECTIVE PARSING

Non-projectivity is a common phenomenon in multilingual dependency parsing. The term non-projectivity indicates that a dependency tree has crossing-arcs, which often appear in morphologically rich languages. Various algorithms have been proposed for both graph-based and transition-based parsing algorithms to produce non-projective trees. For example, the *arc-standard* algorithm (Section 3.1) can be readily extended by adding a *swap* action to handle the non-projectivity, which gives an expected linear and worst-case $O(n^2)$ complexity (Nivre, 2009). Other strategies include the *list-based* algorithm (Nivre, 2008) which is adapted from the Covington algorithm (Covington, 2001), and a further combination of the *list-based* and the *swap-based* algorithm (Choi & McCallum, 2013). Unfortunately, there has been no systematically comparison for these different algorithms in the literature so far.

In this study, however, we focus only on projective parsing because there is no non-projective trees in our source language (English) training data. Consequently, non-projectivities in target languages will not be handled at this moment.[6]

4. Cross-Lingual Word Representation Learning

Prior to introducing our approaches for cross-lingual word representation learning, we briefly review the basic model for learning monolingual word embeddings, which constitutes a subprocedure of the cross-lingual approaches.

4.1 Continuous Bag-of-Words Model

In recent years, various approaches have been studied for learning word embeddings from large-scale plain texts. All approaches are generally derived from the so-called *distributional hypothesis* (Firth, 1957): *"You shall know a word by the company it keeps"*. In this study, we consider the Continuous Bag-of-Words (CBOW) model (Mikolov, Chen, Corrado, & Dean, 2013) as imple-

[4] `https://universaldependencies.github.io/docs/`

[5] `https://github.com/ryanmcd/uni-dep-tb`

[6] Note that for the target languages we address in this paper, non-projectivity is not pervasive. Specifically, the proportion of projective trees presented in their training corpus is respectively 91% for DE, 94% for ES, and 88% for FR.

mented in the open-source toolkit *word2vec*.[7] The basic principle of the CBOW model is to predict each individual word in a sequence given the bag of its context words within a fixed window size as input, using a log-linear classifier. This model avoids the non-linear transformation in hidden layers, and hence can be trained with high efficiency.

With large window size, grouped words using the resulting word embeddings are more topically similar; whereas with small window size, the grouped words will be more syntactically similar (Bansal, Gimpel, & Livescu, 2014). So we set the window size to 1 in our parsing task.

Next, we introduce our approach for inducing bilingual word embeddings. In general, we expect our bilingual word embeddings to preserve translational equivalences. For example, "cooking" (English) should be close to its translation: "kochen" (German) in the embedding space.

4.2 Robust Alignment-Based Projection

Our first method for inducing cross-lingual word embeddings has two stages. First, we learn word embeddings from a source language (S) corpora as in the monolingual case, and then project the monolingual word embeddings to a target language (T), based on word alignments.

Given a sentence-aligned parallel corpus $\mathcal{D}$, we first conduct unsupervised bidirectional word alignment, and then collect an alignment dictionary. Specifically, in each word-aligned sentence pair of $\mathcal{D}$, we keep all alignments with conditional alignment probability exceeding a threshold $\delta = 0.95$ and discard the others. Specifically, let $\mathcal{A}^{T|S} = \{(w_i^T, w_j^S, c_{i,j}), i = 1, 2, ..., N_T; j = 1, 2, ..., N_S\}$ be the alignment dictionary, where $c_{i,j}$ is the number of times when the i^{th} target word w_i^T is aligned to the j^{th} source word w_j^S. N_S and N_T are vocabulary sizes. We use the shorthand $(i, j) \in \mathcal{A}^{T|S}$ to denote a word pair in $\mathcal{A}^{T|S}$. The projection can be formalized as the weighted average of the embeddings of translation words:

$$v(w_i^T) = \sum_{(i,j)\in\mathcal{A}^{T|S}} \frac{c_{i,j}}{c_{i,\cdot}} \cdot v(w_j^S) \tag{1}$$

where $c_{i,\cdot} = \Sigma_j c_{i,j}$, $v(w)$ is the embedding of w.

Obviously, the simple projection method has one drawback: it only assigns word embeddings for those target language words that occur in the word aligned data, which is typically smaller than the monolingual datasets. Therefore, in order to improve the robustness of projection, we utilize a morphology-inspired mechanism, to propagate embeddings from in-vocabulary words to out-of-vocabulary (OOV) words. Specifically, for each OOV word w_{oov}^T, we extract a list of candidate words that is similar to it in terms of *edit distance* (Levenshtein distance), and then set the averaged vector as the embedding of w_{oov}^T. More formally,

$$\begin{aligned} &v(w_{oov}^T) = \underset{w'\in C}{Avg}(v(w')) \\ &\text{where } C = \{w | w \in \mathit{EditDist}(w_{oov}^T, w) \le \tau\} \end{aligned} \tag{2}$$

To reduce noise, we choose a small *edit distance* threshold $\tau = 1$.

The process of robust projection can be viewed as a two-stage graph-propagation algorithm, as illustrated in Figure 3 (left panel). Embeddings are first propagated from source language words to target language words that appear in the bilingual lexicons. Next, monolingual propagation is performed to obtain OOV word embeddings in the target language, using the *edit distance* metric.

[7] http://code.google.com/p/word2vec/

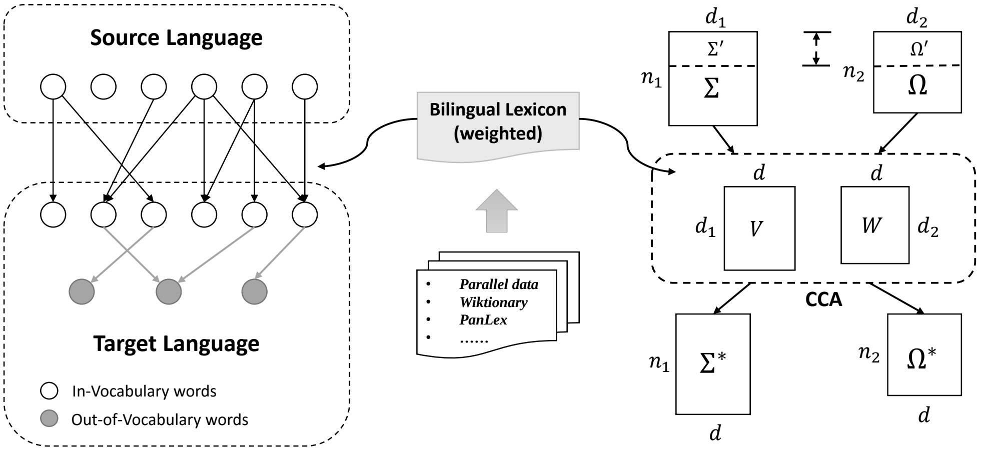

Figure 3: Illustration of robust projection (left) and CCA (right) for inducing cross-lingual word embeddings.

4.3 Canonical Correlation Analysis

The second approach we consider is similar to that of Faruqui and Dyer (2014), which uses CCA to improve monolingual word embeddings with multilingual correlation. CCA is a way of measuring the linear relationship between multidimensional variables. For two multidimensional variables, CCA aims to find two projection matrices to map the original variables to a new basis (lower-dimensional), such that the correlation between the two variables is maximized.

We refer the readers to the work of Hardoon, Szedmak, and Shawe-Taylor (2004) for theoretical foundations and algorithm specifics of CCA. Here let's treat CCA as a black box, and see how CCA can be applied for inducing bilingual word embeddings. Suppose there are already two pre-trained monolingual word embeddings (e.g., English and German): $\Sigma \in \mathbb{R}^{n_1 \times d_1}$ and $\Omega \in \mathbb{R}^{n_2 \times d_2}$. At the first step, we extract a one-to-one alignment dictionary $\mathcal{D} : \Sigma' \leftrightarrow \Omega'$ from the alignment dictionary $\mathcal{A}^{S|T}$.[8] Here, $\Sigma' \subseteq \Sigma$, indicating that every word in Σ' is translated to one word in $\Omega' \subseteq \Omega$, and vice versa.

The process is illustrated in Figure 3 (right panel). Denoting the dimension of resulting word embeddings by $d \leq min(d_1, d_2)$. First, we derive two projection matrices $V \in \mathbb{R}^{d_1 \times d}, W \in \mathbb{R}^{d_2 \times d}$ respectively for Σ' and Ω' using CCA:

$$V, W = CCA(\Sigma', \Omega') \tag{3}$$

Then, V and W are used to project the entire vocabulary Σ and Ω:

$$\Sigma^* = \Sigma V, \quad \Omega^* = \Omega W \tag{4}$$

where $\Sigma^* \in \mathbb{R}^{n_1 \times d}$ and $\Omega^* \in \mathbb{R}^{n_2 \times d}$ are the resulting word embeddings for our cross-lingual task.

[8] $\mathcal{A}^{T|S}$ is also worth trying, but we observed slight performance degradation in our experimental setting.

4.4 Pros and Cons

Contrary to the robust projection approach, CCA assigns embeddings for every word in the monolingual vocabulary. However, one potential limitation is that CCA assumes linear transformation of word embeddings, which is difficult to satisfy. At the mean time, when training the source language parser using the CCA cross-lingual word embeddings, we have to constrained E^w to be fixed, as mentioned in Section 3.1, otherwise, the translational equivalence will be broken. The robust projection approach, however, doesn't have such limitation. Further discussion with experiments will be presented in Section 5.3.2.

Note that both approaches can be generalized to lower-resource languages where parallel bitexts are not available. In that way, the dictionary $\mathcal{A}$ can be readily obtained either using bilingual lexicon induction approaches (Mann & Yarowsky, 2001; Koehn & Knight, 2002; Haghighi, Liang, Berg-Kirkpatrick, & Klein, 2008), or from online-resources like *Wiktionary*[9] and *Panlex*.[10]

5. Experiments

This section describes the experiments. We first describe the data and settings used in the experiments, and then the results.

5.1 Data and Settings

For the pre-training of word embeddings, we use the WMT-2011 monolingual news corpora for English, German and Spanish.[11] For French, we combined the WMT-2011 and WMT-2012 monolingual news corpora.[12] We got the word alignment counts using the *fast-align* toolkit in *cdec* (Dyer et al., 2010) from the parallel news commentary corpora (WMT 2006-10) combined with the Europarl corpus for English–{German, Spanish, French}.[13]

For the training of the neural network dependency parser, we set the number of hidden units to 400. The dimension of embeddings for different features are shown in Table 2.

	Word	POS	Label	Distance	Valency	Cluster
Dim.	50	50	50	5	5	8

Table 2: Dimensions of various types of feature embeddings.

Mini-batch adaptive stochastic gradient descent (AdaGrad) (Duchi, Hazan, & Singer, 2011) is used for optimization. For the CCA approach, we use the implementation of Faruqui and Dyer (2014).

We employ the universal dependency treebanks (UDT v2.0) for a reliable evaluation of our approach for cross-lingual dependency parsing. The universal multilingual treebanks are annotated using the universal POS tagset (Petrov et al., 2012) which contains 12 POS tags, as well as the universal dependencies which defines 40 dependency relations. We follow the standard split of the treebanks for all languages.

[9] https://www.wiktionary.org/

[10] http://panlex.org/

[11] http://www.statmt.org/wmt11/

[12] http://www.statmt.org/wmt12/

[13] http://www.statmt.org/europarl/

5.2 Baseline Systems

We compare our approach with the following systems.

For the first baseline, we evaluate the delexicalized transfer of our neural network-based parser [DELEX], in which we only use the non-lexical features (Figure 2). Here we investigate the effect of the non-local features (*distance*, *valency*). The delexicalized systems which do not include these non-local features is referred to as [DELEX (basic)].

We also compare our approach with the delexicalized parser presented by McDonald et al. (2013) [MCD13], which used a perceptron-based transition-based parser with a beam of size 8, along with richer non-local features (Zhang & Nivre, 2011). Our re-implementation of this approach under the framework of *Zpar* (Zhang & Clark, 2011) is referred to as [MCD13*].

Furthermore, we consider a strong baseline system as proposed by Täckström et al. (2012), which utilized cross-lingual word cluster features to enhance the perceptron-based delexicalized parser [MCD13*+Cluster]. We use the same alignment dictionary as described in Section 4.2 to induce the cross-lingual word clusters. We re-implement the PROJECTED clustering approach described in the work of Täckström et al., which assigns a target word to the cluster with which it is most often aligned:

$$c(w_i^T) = \arg\max_k \sum_{(i,j)\in\mathcal{A}^{T|S}} c_{i,j} \cdot \mathbb{1}[c(w_j^S) = k]$$

Obviously, this method also has the drawback that words that do not occur in the alignment dictionary (OOV) cannot be assigned a cluster. Therefore, we use the same strategy as described in Section 4.2 to find the most likely clusters for the OOV words. Instead of computing the average of embeddings, we solve an `argmax` problem:

$$\begin{aligned} &c(w_{oov}^T) = \arg\max_k \sum_{w'\in\mathcal{C}} \mathbb{1}[c(w') = k] \\ &\text{where } \mathcal{C} = \{w | \mathit{EditDist}(w_{oov}^T, w) \le \tau\} \end{aligned} \tag{5}$$

τ is set to 1 constantly. Instead of the clustering model of Uszkoreit and Brants (2008), we use Brown clustering (1992) to induce hierarchical word clusters, where each word is represented as a bit-string. We use the same word cluster feature templates from Täckström et al. (2012), and set the number of Brown clusters to 256.

5.3 Experimental Results

All of the parsing models are trained using the development data from English for early-stopping. Table 3 lists the results of the cross-lingual transfer experiments for dependency parsing. Table 4 further summarizes each of the experimental gains detailed in Table 3.

We first examine the benefit brought by the non-local *distance* and *valency* features. As observed in the comparison of DELEX (basic) and DELEX, marginal improvements are obtained for DE and FR, and more significant improvements for ES. Therefore, we adopted these features in all of the following experiments.

Our delexicalized system obtains slightly lower performance than those reported by McDonald et al. (2013) (MCD13), because we used greedy decoding and local training. Our re-implementation of McDonald et al.'s work attains comparable performance with MCD13. For all languages we consider in this study, by using cross-lingual word embeddings either from alignment-based projection or CCA, we obtain statistically significant improvements against the delexicalized system, both in

	Unlabeled Attachment Score (UAS)					Labeled Attachment Score (LAS)				
	EN	DE	ES	FR	AVG	EN	DE	ES	FR	AVG
Delex (basic)	83.63	56.85	67.28	68.70	*64.28*	79.37	47.06	56.43	57.73	*53.74*
Delex	83.67	57.01	68.05	68.85	*64.64*	79.42	47.12	56.99	57.78	*53.96*
Proj	91.96	60.07	71.42	71.36	*67.62*	90.48	49.94	61.76	61.55	*57.75*
Proj+Cluster	92.33	60.35	**71.90**	**72.93**	***68.39***	90.91	**51.54**	**62.28**	**63.12**	***58.98***
CCA	90.62†	59.42	68.87	69.58	*65.96*	88.88†	49.32	59.65	59.50	*56.16*
CCA+Cluster	92.03†	**60.66**	71.33	70.87	*67.62*	90.49†	51.29	61.69	61.50	*58.16*
McD13	83.33	58.50	68.07	70.14	*65.57*	78.54	48.11	56.86	58.20	*54.39*
McD13*	84.44	57.30	68.15	69.91	*65.12*	80.30	47.34	57.12	58.80	*54.42*
McD13*+Cluster	90.21	60.55	70.43	72.01	*67.66*	88.28	50.20	60.96	61.96	*57.71*

Table 3: Cross-lingual transfer dependency parsing from English on the test dataset of 4 universal multilingual treebanks. Results measured by unlabeled attachment score (UAS) and labeled attachment score (LAS). Delex (basic) is the delexicalized model without non-local features (*distance*, *valency*). * denotes our re-implementation of McD13. Since the model varies for different target languages in the CCA-based approach, † indicates the averaged UAS/LAS.

Experimental Contribution			DE/ES/FR Avg. (Relative)
Proj	*vs.*	Delex	*+3.79 (8.2%)*
CCA	*vs.*	Delex	*+2.19 (4.8%)*
Proj	*vs.*	McD13*	*+3.33 (7.3%)*
CCA	*vs.*	McD13*	*+1.74 (3.8%)*
Proj+Cluster	*vs.*	Proj	*+1.23 (2.9%)*
CCA+Cluster	*vs.*	CCA	*+2.00 (4.6%)*
McD13*+Cluster	*vs.*	McD13*	*+3.29 (7.2%)*
Proj+Cluster	*vs.*	Delex	*+5.02 (10.9%)*
CCA+Cluster	*vs.*	Delex	*+4.20 (9.1%)*
Proj+Cluster	*vs.*	McD13*	*+4.46 (9.8%)*
CCA+Cluster	*vs.*	McD13*	*+3.74 (8.2%)*
Proj+Cluster	*vs.*	McD13*+Cluster	*+1.27 (3.0%)*
CCA+Cluster	*vs.*	McD13*+Cluster	*+0.45 (1.1%)*

Table 4: Summary of each of the experimental gains detailed in Table 3, in both absolute LAS gain and relative error reduction. All gains are statistically significant using MaltEval (Nilsson & Nivre, 2008) at $p < 0.01$.

UAS and LAS. Interestingly, we notice that Proj consistently outperforms CCA by a significant margin, and is comparable to McD13*+Cluster. Further analysis to this observation will be conducted in Section 5.3.1 and 5.3.2.

Type	Feature Templates
Cluster	$E^c_{S_i}, E^c_{B_i}, i = 0,1,2$ $E^c_{lc1(S_i)}, E^c_{rc1(S_i)}, E^c_{lc2(S_i)}, E^c_{rc2(S_i)}, i = 0,1$ $E^c_{lc1(lc1(S_i))}, E^c_{rc1(rc1(S_i))}, i = 0,1$

Table 5: Word cluster feature templates.

Our framework is flexible for incorporating richer features simply by embedding them into continuous vectors. Thus we further embed the cross-lingual word cluster features into our model, together with the proposed cross-lingual word embeddings. The cluster feature templates are shown in Table 5, which is similar to the POS tag feature templates. As shown in Table 3, significant additive improvements are obtained for both PROJ and CCA by embedding the cluster features. Compared with our delexicalized system, the relative error is reduced by up to 13.1% in UAS, and up to 12.6% in LAS. The combined system further outperforms MCD13*+Cluster significantly .

5.3.1 EFFECT OF ROBUST PROJECTION

Since in both PROJ and the induction of cross-lingual word clusters, we use *edit distance* measure for OOV words, we would like to see how this affects the performance of parsing.

Intuitively, higher coverage of projected words in the test dataset should promote the parsing performance more. To verify this, we further conduct experiments under both settings using the PROJ+Cluster model. For robust projection, we examine the effect of edit distances ranging from 1 to 3. Results are shown in Table 6. Improvements are observed for all languages when using robust projection with *edit distance* measure, especially for FR, where the highest coverage gain is obtained by robust projection. We also observe slightly improvements for DE and ES when using an edit distance of 2. But performance starts to degrade when it gets larger. This is reasonable, since larger edit distance increases the word coverage, but also introduces more noise.

		Simple	Robust		
			τ=1	*τ=2*	*τ=3*
DE	coverage	91.37	94.70	96.50	97.47
	UAS	59.74	60.35	**60.53**	**60.53**
	LAS	50.84	51.54	**51.70**	51.69
ES	coverage	94.51	96.67	97.75	98.47
	UAS	70.97	71.90	**72.00**	71.93
	LAS	61.34	62.28	**62.34**	62.27
FR	coverage	90.83	97.60	98.33	98.58
	UAS	71.17	**72.93**	72.79	72.70
	LAS	61.72	**63.12**	63.02	62.94

Table 6: Effect of robust projection.

5.3.2 EFFECT OF FINE-TUNING WORD EMBEDDINGS

Another reason for the effectiveness of PROJ over CCA lies in the fine-tuning of word embeddings while training the parser.

CCA can be viewed as a joint method for inducing cross-lingual word embeddings. When training the source language dependency parser with cross-lingual word embeddings derived from CCA, the EN word embeddings should be fixed. Otherwise, the translational equivalence will be broken. However, for PROJ, there is no such limitation. Word embeddings can be updated as other non-lexical feature embeddings, in order to obtain a more accurate dependency parser. We refer to this procedure as a *fine-tuning* process to the word embeddings. To verify the benefits of *fine-tuning*, we conduct experiments to see relative loss if word embeddings are fixed while training. Results are shown in Table 7, which indicates that *fine-tuning* indeed offers considerable help.

		Fixed	Fine-tuning	Δ
DE	UAS	59.74	**60.07**	+0.33
	LAS	49.44	**49.94**	+0.50
ES	UAS	70.10	**71.42**	+1.32
	LAS	61.31	**61.76**	+0.45
FR	UAS	70.65	**71.36**	+0.71
	LAS	60.69	**61.50**	+0.81

Table 7: Effect of fine-tuning word embeddings.

5.4 Compare with Existing Bilingual Word Embeddings

In this section, we compare our bilingual embeddings with several previous approaches in the context of dependency parsing. To the best of our knowledge, this is the first work on evaluation of bilingual word embeddings in syntactic tasks.

The approaches we consider include the multi-task learning approach (Klementiev et al., 2012) [MTL], the bilingual auto-encoder approach (Chandar A P et al., 2014) [BIAE], the bilingual compositional vector model (Hermann & Blunsom, 2014) [BICVM], and the bilingual bag-of-words approach (Gouws et al., 2015) [BILBOWA].

For MTL and BIAE, we adopt their released word embeddings directly due to the inefficiency of training.[14] For BICVM and BILBOWA, we re-run their systems on the same dataset as our previous experiments.[15] Results are summarized in Table 8.

CCA and PROJ consistently outperforms all other approaches in all languages, and PROJ performs the best. The inferior performance of MTL and BIAE is partly due to the low word coverage. For example, they cover only 31% of words in the universal DE test treebank, whereas the CCA and PROJ covers over 70%. Moreover, BIAE, BICVM and BILBOWA introduce sentence-level translational equivalence as objectives or regularizers for learning bilingual word embeddings. These approaches are advantageous in that they don't assume/require word alignment. However, word-to-word translational equivalence cannot be well preserved in this way.

[14] The MTL embeddings are normalized before training.

[15] BICVM only uses the bilingual parallel dataset.

	DE		ES		FR	
	UAS	LAS	UAS	LAS	UAS	LAS
MTL (Klementiev et al., 2012)‡	56.93	46.22	67.71	58.43	67.51	57.27
BIAE (Chandar A P et al., 2014)‡	53.74	43.68	58.81	46.66	60.10	49.47
BICVM (Hermann & Blunsom, 2014)	56.30	46.99	67.78	58.08	69.13	58.13
BILBOWA (Gouws et al., 2015)	54.51	44.95	67.23	56.16	64.82	52.73
CCA	59.42	49.32	68.87	59.65	69.58	59.50
PROJ	**60.07**	**49.94**	**71.42**	**61.76**	**71.36**	**61.55**

Table 8: Comparison with existing bilingual word embeddings. ‡For MTL and BIAE, we use their released bilingual word embeddings.

Target Word (ES)	Neighboring Words (EN)					
	PROJ	CCA	MTL	BIAE	BICVM	BILBOWA
china (china)	india russia taiwan chinese	russia indonesia beijing chinese	china independent sumitomo malaysian	korea india chinese brazil	chinese chinois sino 33.55	helsinki bulgarians constituting market
problemas (problems)	problem difficulties troubles issues	problems woes troubles dilemmas	events sanctions conditions laws	problem greatly highlighted scale	problematic problematical difficulties troubles	deficiencies situations omissions attentively
septiembre (september)	october august january december	december july october june	december february july november	month april scheduled march	11th 11.00 11 eleventh	a.m p.m twelve 1998-1999

Table 9: Target words in Spanish and their 4 most similar words in English, as induced by various approaches.

To verify this assumption, we taking EN/ES as a case study. We manually inspect the 4 most similar words (by cosine similarity) in English to a given set of words in Spanish (Table 9). We can observe both semantic and syntactic shifting in the k-nearest neighbors prediction of BIAE, BICVM and BILBOWA, whereas PROJ and CCA give more translational equivalent predictions. For example, BICVM yields adjective like *problematical* for the target noun *problemas*; BILBOWA yields semantic-related word *market* for *china*. In general, PROJ is the most robust approach, behaving consistently well for most of the sampled words.

It is worth noting that we don't assume/require bilingual parallel data in CCA and PROJ. What we need in practice is a bilingual lexicon for each paired languages. This is especially important for generalizing our approaches to lower-resource languages, where parallel texts are not available.

6. Target-Language Adaptation with Minimal Supervision

It is important for us to distinguish what linguistic structures can be learned via cross-lingual transfer versus what can only be learned on the basis of monolingual information in the language to be parsed. Intuitively, cross-lingual approaches can only learn the common dependency structures shared between the source and target language. However, for many languages, there are some specialized (language-specific) syntactic characteristics that are can only be learned from data in the target language.

Take the *adjective-noun* order for example, in Spanish and French, adjectives often appears after the nouns, thus forming a right-directed arc labeled by *amod*, whereas in English, the *amod* (adjectival modifier) arcs are mostly left-directed, as illustrated in Figure 4. Another example is the *subject-verb-object* order. In German, verbs often appear at the end of a sentence in V2 position, which causes much more left-directed *dobj* (direct object) arcs than in English (Figure 5). These differences can be clearly observed from the universal treebanks. Table 10 shows the significant distribution divergence between left-directed and right-directed arcs of *dobj* and *amod* relations in treebanks from different languages.

Relation: *dobj*; Language: EN *vs.* DE

	$dobj_{\curvearrowright}$	$dobj_{\curvearrowleft}$	ratio
EN	38,395	764	50.3 : 1
DE	4,277	3,457	1.2 : 1

Relation: *amod*; Language: EN *vs.* ES, FR

	$amod_{\curvearrowright}$	$amod_{\curvearrowleft}$	ratio
EN	1,667	57,864	1 : 34.7
ES	14,876	5,205	2.9 : 1
FR	12,919	4,910	2.6 : 1

Table 10: Distribution divergences of left-directed and right-directed arcs with *dobj* relation in EN and DE (top), and *amod* relation in EN and ES/FR (bottom).

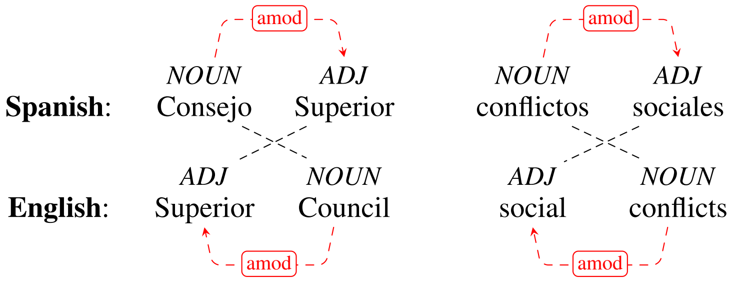

Figure 4: Reverse direction of the *amod* relation in Spanish and English. French also has the adjectives following the nouns.

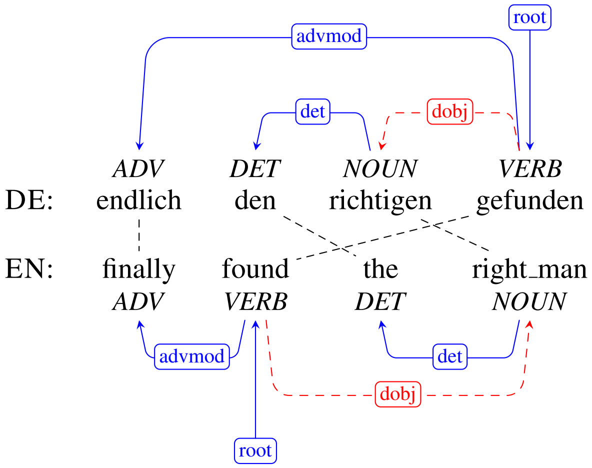

Figure 5: Reverse direction of the *dobj* relation in German and English.

Therefore, in this section, we investigate how much our cross-lingual transfer model can be improved by annotating a small amount of labeled training data at target language side. Even though building large-scale treebanks of low-resource languages for supervised learning is costly, annotating dependency structures for a small amount of sentences (e.g., 100) is not that difficult.

We still conduct experiments on the universal dependency treebanks, which provide labeled training data for multiple languages. For each language we studied (DE, ES, FR), we incrementally augment the amount of labeled sentences from 100 to 1,000 with a step of 100, to adapt the parameters of the cross-lingual transfer model to the specific target language. Theoretically, since target language treebanks contain non-projective trees, it would make more sense to apply non-projective algorithms (e.g., *swap-based*) for target language adaptation. In this way, however, $\mathbf{W}_2$ has to be re-trained from scratch, which doesn't show good performance in our experiments since the minimally supervised data is very small. Consequently, we still rely on the *arc-standard* algorithm for adaption. The process is almost the same as training the source language parser as described in Section 3, except that the word embedding matrix E^w is fixed, while the rest of parameters in θ ($E^{\{t,l,d,v,c\}}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1$) are optimized using the augmented labeled data from the target language, taking Equation 3.1 as objective function. No development data is used during this process, thus we simply perform parameter updating for 2,000 iterations.

In addition, we built another strong baseline system which employs the same augmented labeled training data for supervised learning. In this system, we utilize both word embeddings and Brown clusters as features, which are derived separately for each language.

As shown in Figure 6, the results are really promising. The PROJ+Cluster and CCA+Cluster systems consistently outperform the delexicalized system and the supervised system by a significant margin. PROJ+Cluster and CCA+Cluster in general achieve comparable performances, while CCA+Cluster is slightly better.

It is worthy noting that the performances of PROJ+Cluster and CCA+Cluster are boosted by augmenting only 100 sentences. Take DE for example, UAS is increased from 60.35% to 68.91%, and LAS from 51.54% to 61.54%, which is nearly equal to the effect of using 1,000 sentences for supervised learning. This observation demonstrates the great potential of our cross-lingual transfer system for practical usage.

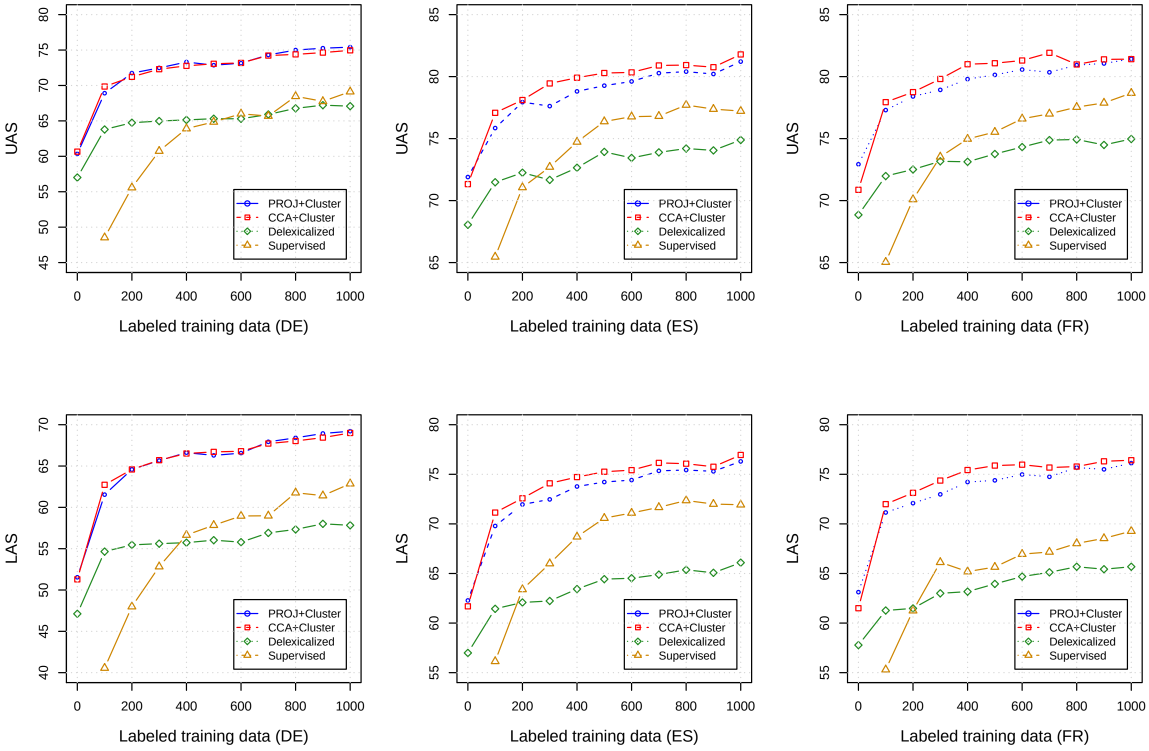

Figure 6: Target-language adaptation by incrementally augmenting labeled training data (sentences) to fine-tune the cross-lingual transfer model. Performances are evaluated using UAS (top) and LAS (bottom). Note that the points whose x coordinates are 0 represent our cross-lingual transfer performance, where no labeled training data are used.

Analysis. Our primary hypothesis is that by incorporating data in the target language, our model can be able to learn the special syntactic patterns that are not consistent with the source language. To verify this, we further study the influence of target-language adaptation on the two special relations: *dobj* (DE) and *amod* (ES, FR), by measuring their *precision* and *recall* changes with the use of only 100 target language sentences. Results are shown respectively in Table 11 and Table 12. We observe great improvements in recall for these relations, which indicates that the model indeed gains the ability of learning target-language-specific dependency structures with the supervision of only 100 sentences.

7. Related Studies

The cross-lingual annotation projection method was pioneered by Yarowsky, Ngai, and Wicentowski (2001) for shallow NLP tasks (POS tagging, NER, etc.), and later applied to dependency parsing (Hwa et al., 2005; Smith & Eisner, 2009; Zhao et al., 2009; Jiang et al., 2011; Tiedemann, 2014). Most work along this line has been dedicated to improving the robustness of syntactic pro-

Relation: *dobj*; Language: DE

	Precision	Recall
PROJ+Cluster	41.45	31.09
+100	41.90	51.40
Δ	↑ 0.45	↑ **20.31**
CCA+Cluster	39.47	31.74
+100	43.59	57.57
Δ	↑ 4.12	↑ **25.83**

Table 11: Effect of minimal supervision (100 sentences) on *dobj*.

Relation: *amod*; Language: ES, FR

	ES		FR	
	Precision	Recall	Precision	Recall
PROJ+Cluster	94.97	80.05	92.94	81.70
+100	91.60	92.52	93.61	95.75
Δ	↓ 3.37	↑ **12.47**	↑ 0.67	↑ **14.05**
CCA+Cluster	93.37	77.31	92.08	72.22
+100	91.85	92.77	92.77	96.41
Δ	↓ 1.52	↑ **15.46**	↑ 0.69	↑ **24.19**

Table 12: Effect of minimal supervision (100 sentences) on *amod*.

jection and alleviating the noise and errors introduced by word alignment-based projection. Typical approaches include *soft projection* (Li, Zhang, & Chen, 2014), *treebank translation* (Tiedemann, Agić, & Nivre, 2014), *distribution transfer* (Ma & Xia, 2014), and the most recently proposed *density-driven projection* (Rasooli & Collins, 2015). It is worth mentioning that remarkable results have been achieved through annotation projection methods (Tiedemann, 2015; Rasooli & Collins, 2015), due in large part to that parsers are trained at the target language side.

For cross-lingual model transfer, learning cross-lingual feature representations has been a promising direction. Typical approaches include cross-lingual word clustering (Täckström et al., 2012) which is employed in this paper as a baseline system, and projection features (Durrett, Pauls, & Klein, 2012). Kozhevnikov and Titov (2014) derived a linear projection that maps target instances to source-side feature representations, which to some extent is similar to our CCA approach. Xiao and Guo (2014) learned cross-lingual word embeddings and applied them to MSTParser for linguistic transfer, which inspired our work. Søgaard et al. (2015) obtained multi-source unified word embeddings via inverted indexing in Wikipedia, and applied them to various NLP tasks. However, their results didn't show significant improvements in parsing. Nevertheless, the idea of utilizing multi-source information for learning cross-lingual word embeddings makes great sense. More recently, Duong et al. (2015a, 2015b) also utilized the neural network architecture with parameter sharing between parsers of different languages. However, their approach requires annotated treebanks from the target language side, which makes it distinct from our transfer parsing framework. In addition to representation learning, attempts were also made to integrate monolingual linguistic features into the parsing models, such as manually constructed universal dependency parsing rules (Naseem, Chen, Barzilay, & Johnson, 2010) and manually specified typological features (Naseem, Barzilay, & Globerson, 2012; Zhang & Barzilay, 2015).

Using neural networks for dependency parsing is not a new approach. To the best of our knowledge, Mayberry and Miikkulainen (1999) presented the first work that explored neural networks for shift-reduce constituent-based parsing. They used one-hot feature representations. Henderson (2004) used a simple synchrony network to predict parse decisions in a constituency parser, and was the first to use neural networks in a broad-coverage Penn Treebank parser. Titov and Henderson (2007) applied Incremental Sigmoid Belief Networks to constituent-based parsing. Garg and Henderson (2011) later extended this work to transition-based dependency parsing using a Temporal Restricted Boltzman Machine. These parsers, however, are much less scalable in practice. Earlier progress made in using deep learning for parsing includes work by Collobert (2011) and Socher et al. (2013) for constituent-based parsing, and Stenetorp (2013) who built recursive neural networks for transition-based dependency parsing.

8. Conclusion

This paper proposes a novel framework based on distributed representations for cross-lingual dependency parsing. Two algorithms are proposed for the induction of cross-lingual word representations, namely robust projection and CCA, which bridge the *lexical feature gap*.

Experiments show that by using cross-lingual word embeddings derived from either approach, the transferred parsing performance can be improved significantly against the delexicalized system. A notable observation is that our projection method performs significantly better than CCA. Additionally, our framework is flexibly able to incorporate the cross-lingual word cluster features, with further significant gains in each use. The combined system significantly outperforms the delexicalized systems on all languages, by an average of 10.9% error reduction on LAS, and further significantly outperforms the models of McDonald et al. (2013) augmented with projected word cluster features.

Furthermore, we show that the performance of our cross-lingual transfer system on a specific target language can be boosted by minimal supervision from that language, which is of great significance for practical usage.

Acknowledgments

We are grateful to Manaal Faruqui for providing the bilingual resources. We thank Ryan McDonald for pointing out the evaluation issue in the experiment. We also thank Sharon Busching for the proofreading and the anonymous reviewers for the insightful comments and suggestions. This work was supported by the National Key Basic Research Program of China via grant 2014CB340503 and the National Natural Science Foundation of China (NSFC) via grant 61133012 and 61370164. Corresponding author: Wanxiang Che, E-mail: car@ir.hit.edu.cn.

References

Bansal, M., Gimpel, K., & Livescu, K. (2014). Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computa-*

tional Linguistics (Volume 2: Short Papers), pp. 809–815, Baltimore, Maryland. Association for Computational Linguistics.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, *18*(4), 467–479.

Buchholz, S., & Marsi, E. (2006). Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pp. 149–164, New York City. Association for Computational Linguistics.

Cao, Y., & Khudanpur, S. (2014). Online learning in tensor space. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 666–675, Baltimore, Maryland. Association for Computational Linguistics.

Carreras, X. (2007). Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pp. 957–961, Prague, Czech Republic. Association for Computational Linguistics.

Chandar A P, S., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., & Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems 27*, pp. 1853–1861. Curran Associates, Inc.

Chen, D., & Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 740–750, Doha, Qatar. Association for Computational Linguistics.

Choi, J. D., & McCallum, A. (2013). Transition-based dependency parsing with selectional branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1052–1062, Sofia, Bulgaria. Association for Computational Linguistics.

Collobert, R. (2011). Deep learning for efficient discriminative parsing. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 224–232, Fort Lauderdale, FL, USA. JMLR.org.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 160–167, Helsinki, Finland. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Covington, M. A. (2001). A fundamental algorithm for dependency parsing. In *Proceedings of the 39th annual ACM southeast conference*, pp. 95–102.

De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pp. 449–454, Genoa, Italy. European Language Resources Association (ELRA).

De Marneffe, M.-C., & Manning, C. D. (2008). The stanford typed dependencies representation. In *COLING 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pp. 1–8, Manchester, UK. Association for Computational Linguistics.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.

Duong, L., Cohn, T., Bird, S., & Cook, P. (2015a). Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. 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)*, pp. 845–850, Beijing, China. Association for Computational Linguistics.

Duong, L., Cohn, T., Bird, S., & Cook, P. (2015b). A neural network model for low-resource universal dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 339–348, Lisbon, Portugal. Association for Computational Linguistics.

Durrett, G., Pauls, A., & Klein, D. (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*, pp. 1–11, Jeju Island, Korea. Association for Computational Linguistics.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., & Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. 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)*, pp. 334–343, Beijing, China. Association for Computational Linguistics.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., & Resnik, P. (2010). cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, pp. 7–12, Uppsala, Sweden. Association for Computational Linguistics.

Eisner, J. M. (1996). Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pp. 340–345, Copenhagen, Denmark. Association for Computational Linguistics.

Faruqui, M., & Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 462–471, Gothenburg, Sweden. Association for Computational Linguistics.

Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. In *Studies in linguistic analysis*, pp. 1–32. Blackwell.

Garg, N., & Henderson, J. (2011). Temporal restricted boltzmann machines for dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 11–17, Portland, Oregon, USA. Association for Computational Linguistics.

Gouws, S., Bengio, Y., & Corrado, G. (2015). Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 748–756, Lille, France.

Guo, J., Che, W., Wang, H., & Liu, T. (2014). Revisiting embedding features for simple semi-supervised learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pp. 110–120, Doha, Qatar. Association for Computational Linguistics.

Guo, J., Che, W., Yarowsky, D., Wang, H., & Liu, T. (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)*, pp. 1234–1244, Beijing, China. Association for Computational Linguistics.

Haghighi, A., Liang, P., Berg-Kirkpatrick, T., & Klein, D. (2008). Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pp. 771–779, Columbus, Ohio. Association for Computational Linguistics.

Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, *16*(12), 2639–2664.

Henderson, J. (2004). Discriminative training of a neural network statistical parser. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pp. 95–102, Barcelona, Spain.

Hermann, K. M., & Blunsom, P. (2014). Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 58–68, Baltimore, Maryland. Association for Computational Linguistics.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., & Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, *11*(03), 311–325.

Jiang, W., Liu, Q., & Lv, Y. (2011). Relaxed cross-lingual projection of constituent syntax. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1201, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p-p. 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Klein, D., & Manning, C. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pp. 478–485, Barcelona, Spain.

Klementiev, A., Titov, I., & Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pp. 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

Koehn, P., & Knight, K. (2002). Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pp. 9–16, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, *51*(3), 455–500.

Koo, T., & Collins, M. (2010). Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1–11, Uppsala, Sweden. Association for Computational Linguistics.

Kozhevnikov, M., & Titov, I. (2014). Cross-lingual model transfer using feature representation projection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 579–585, Baltimore, Maryland. Association for Computational Linguistics.

Lei, T., Xin, Y., Zhang, Y., Barzilay, R., & Jaakkola, T. (2014). Low-rank tensors for scoring dependency structures. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1381–1391, Baltimore, Maryland. Association for Computational Linguistics.

Li, Z., Zhang, M., & Chen, W. (2014). Soft cross-lingual syntax projection for dependency parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 783–793, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Ma, X., & Xia, F. (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)*, pp. 1337–1348, Baltimore, Maryland. Association for Computational Linguistics.

Mann, G. S., & Yarowsky, D. (2001). Multipath translation lexicon induction via bridge languages. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, NAACL '01, pp. 1–8, Pittsburgh, Pennsylvania. Association for Computational Linguistics.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, *19*(2), 313–330.

Mayberry, M. R., & Miikkulainen, R. (1999). Sardsrn: a neural network shift-reduce parser. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 820–827. Morgan Kaufmann Publishers Inc.

McDonald, R., Crammer, K., & Pereira, F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 91–98, Ann Arbor, Michigan. Association for Computational Linguistics.

McDonald, R., & Nivre, J. (2007). Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 122–131, Prague, Czech Republic. Association for Computational Linguistics.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., & Lee, J. (2013). Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

McDonald, R., Petrov, S., & Hall, K. (2011). Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

McDonald, R. T., & Pereira, F. C. (2006). Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics*, pp. 81–88, Trento, Italy. The Association for Computer Linguistics.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *International Conference on Learning Representations (ICLR) Workshop*.

Naseem, T., Barzilay, R., & Globerson, A. (2012). Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 629–637, Jeju Island, Korea. Association for Computational Linguistics.

Naseem, T., Chen, H., Barzilay, R., & Johnson, M. (2010). Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1234–1244, Cambridge, MA. Association for Computational Linguistics.

Nilsson, J., & Nivre, J. (2008). Malteval: an evaluation and visualization tool for dependency parsing.. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pp. 161–166, Marrakech, Morocco. European Language Resources Association (ELRA).

Nivre, J. (2003). An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pp. 149–160, Nancy, France. Association for Computational Linguistics.

Nivre, J. (2004). Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pp. 50–57, Barcelona, Spain. Association for Computational Linguistics.

Nivre, J. (2008). Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, *34*(4), 513–553.

Nivre, J. (2009). Non-projective dependency parsing in expected linear time. 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*, pp. 351–359, Suntec, Singapore. Association for Computational Linguistics.

Nivre, J., Hall, J., & Nilsson, J. (2004). Memory-based dependency parsing. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pp. 49–56, Boston, Massachusetts, USA. Association for Computational Linguistics.

Petrov, S., Das, D., & McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pp. 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Rasooli, M. S., & Collins, M. (2015). Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 328–338, Lisbon, Portugal. Association for Computational Linguistics.

Smith, D. A., & Eisner, J. (2009). Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 822–831, Singapore. Association for Computational Linguistics.

Socher, R., Bauer, J., Manning, C. D., & Andrew Y., N. (2013). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 455–465, Sofia, Bulgaria. Association for Computational Linguistics.

Søgaard, A., Agić, v., Martínez Alonso, H., Plank, B., Bohnet, B., & Johannsen, A. (2015). Inverted indexing for cross-lingual nlp. 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)*, pp. 1713–1722, Beijing, China. Association for Computational Linguistics.

Srivastava, N., & Salakhutdinov, R. R. (2012). Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems 25*, pp. 2222–2230. Curran Associates, Inc.

Stenetorp, P. (2013). Transition-based dependency parsing using recursive neural networks. In *Deep Learning Workshop at NIPS*, Lake Tahoe, Nevada, USA.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. Curran Associates, Inc.

Täckström, O., McDonald, R., & Uszkoreit, J. (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*, pp. 477–487, Montréal, Canada. Association for Computational Linguistics.

Tiedemann, J. (2014). Rediscovering annotation projection for cross-lingual parser induction. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 1854–1864, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Tiedemann, J. (2015). Cross-lingual dependency parsing with universal dependencies and predicted PoS labels., 340–349.

Tiedemann, J., Agić, v., & Nivre, J. (2014). Treebank translation for cross-lingual parser induction., 130–140.

Titov, I., & Henderson, J. (2007). Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pp. 947–951, Prague, Czech Republic. Association for Computational Linguistics.

Turian, J., Ratinov, L.-A., & Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 384–394, Uppsala, Sweden. Association for Computational Linguistics.

Uszkoreit, J., & Brants, T. (2008). Distributed word clustering for large scale class-based language modeling in machine translation. In *Proceedings of ACL-08: HLT*, pp. 755–762, Columbus, Ohio. Association for Computational Linguistics.

Wang, M., & Manning, C. D. (2013). Effect of non-linear deep architecture in sequence labeling. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 1285–1291, Nagoya, Japan. Asian Federation of Natural Language Processing.

Weiss, D., Alberti, C., Collins, M., & Petrov, S. (2015). Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 323–333, Beijing, China. Association for Computational Linguistics.

Xiao, M., & Guo, Y. (2014). Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pp. 119–129, Ann Arbor, Michigan. Association for Computational Linguistics.

Xue, N., Xia, F., Chiou, F.-D., & Palmer, M. (2005). The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, *11*(02), 207–238.

Yamada, H., & Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pp. 195–206, Nancy, France. Association for Computational Linguistics.

Yarowsky, D., Ngai, G., & Wicentowski, R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pp. 1–8, San Diego, CA, USA. Association for Computational Linguistics.

Zeman, D., Dušek, O., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtsky̌, Z., & Hajič, J. (2014). Hamledt: Harmonized multi-language dependency treebank. *Language Resources and Evaluation*, *48*(4), 601–637.

Zhang, Y., & Barzilay, R. (2015). Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1857–1867, Lisbon, Portugal. Association for Computational Linguistics.

Zhang, Y., & Clark, S. (2011). Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, *37*(1), 105–151.

Zhang, Y., & Nivre, J. (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*, pp. 188–193, Portland, Oregon, USA. Association for Computational Linguistics.

Zhao, H., Song, Y., Kit, C., & Zhou, G. (2009). Cross language dependency parsing using a bilingual lexicon. 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*, pp. 55–63, Suntec, Singapore. Association for Computational Linguistics.

Zhou, H., Zhang, Y., Huang, S., & Chen, J. (2015). A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1213–1222, Beijing, China. Association for Computational Linguistics.