Cross-lingual Dependency Parsing Based on Distributed Representations

Jiang Guo[1]*, **Wanxiang Che**[1], **David Yarowsky**[2], **Haifeng Wang**[3], **Ting Liu**[1]
[1]Center for Social Computing and Information Retrieval, Harbin Institute of Technology
[2]Center for Language and Speech Processing, Johns Hopkins University
[3]Baidu Inc., Beijing, China
{jguo, car, tliu}@ir.hit.edu.cn
yarowsky@jhu.edu, wanghaifeng@baidu.com

Abstract

This paper investigates the problem of cross-lingual dependency parsing, aiming at inducing dependency parsers for low-resource languages while using only training data from a resource-rich language (e.g. English). Existing 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 able 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 McDonald et al. (2013) augmented with projected cluster features on identical data.

1 Introduction

Dependency Parsing has been one of NLP's long-standing central problems. The majority of work on dependency parsing has been dedicated to resource-rich languages, such as English and Chinese. For these languages, there exist large-scale annotated treebanks that can be used for supervised training of dependency parsers. However, for most of the languages in the world, there are few or even no labeled training data for parsing, and it is labor intensive and time-consuming to manually build treebanks for all languages. This fact has given rise to a number of research on unsupervised methods (Klein and Manning, 2004), annotation projection methods (Hwa et al., 2005), and model transfer methods (McDonald et al., 2011) for predicting linguistic structures. In this study, we focus on the model transfer methods, which attempt to build parsers for low-resource languages by exploiting treebanks from resource-rich languages.

The major obstacle in transferring a parsing system from one language to another is the lexical features, e.g. words, which are not directly transferable across languages. To solve this problem, McDonald et al. (2011) build 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-type dependencies (De Marneffe et al., 2006; De Marneffe and Manning, 2008), these non-lexical features are not predictive enough. Täckström et al. (2012) propose 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 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

*This work was done while the author was visiting JHU.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1234–1244,
Beijing, China, July 26-31, 2015.

the discrete lexical features 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 and Manning, 2013). Chen and Manning (2014) propose a transition-based dependency parser using a neural network architecture, which is simple but works well on several 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 robust projection, and the second approach is based on canonical correlation analysis. Both approaches are simple to implement and are scalable to large data.

We evaluate our model on the universal multilingual treebanks (McDonald et al., 2013). Case studies include transferring from English to German, Spanish and French. 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 McDonald et al. (2013) augmented with cluster features on identical data. The original major contributions of this paper include:

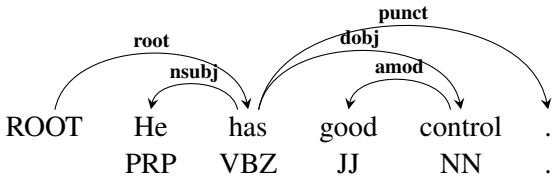

Figure 1: An example labeled dependency tree.

- 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 representations, that bridge the *lexical feature gap* in cross-lingual dependency parsing.

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

2 Background

2.1 Dependency Parsing

Given an input sentence $\mathbf{x} = w_0 w_1 ... w_n$, the goal of dependency parsing is to build a dependency tree (Figure 1), which can be denoted by $\mathbf{d} = \{(h, m, l) : 0 \leq h \leq n; 0 < m \leq n, l \in \mathcal{L}\}$. (h, m, l) indicates a directed arc from the head word w_h to the modifier w_m with a dependency label l, and $\mathcal{L}$ is the label set. The mainstream models that have been proposed for dependency parsing can be described as either graph-based models or transition-based models (McDonald and Nivre, 2007).

Graph-based models 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 structures (e.g. arcs) depending on the order of a model. Transition-based models aim to predict a transition sequence from an initial parser state to some terminal states, depending on the parsing history. This approach has a lot of interest since it is fast (linear time) and can incorporate rich non-local features (Zhang and Nivre, 2011).

It has been considered that simple transition-based parsing using greedy decoding and local training is not as accurate as graph-based parsers or transition-based parsers with beam-search and

global training (Zhang and Clark, 2011). Recently, Chen and Manning (2014) show that greedy transition-based parsers can be greatly improved by using 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. In this study, we choose Chen and Manning's architecture to build our basic dependency parsing model.

2.2 Distributed Representations for NLP

In recent years, there has been a trend in the NLP research community of learning distributed representations for different natural language units, from morphemes, words and phrases, to sentences and documents. Using distributed representations, these symbolic units are embedded into a low-dimensional and continuous space, thus it is often referred to as *embeddings*.[1]

In general, there are two major ways of applying distributed representations to NLP tasks. First, they can be fed into existing supervised NLP systems as augmented features in a semi-supervised manner. This kind of approach has been adopted in a variety of applications (Turian et al., 2010). Despite its 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 existing NLP systems (Wang and Manning, 2013). One remedy is to discretize the distributed feature representations, as studied in Guo et al. (2014). However, we believe that a non-linear system, e.g. a neural network, is a more powerful and effective solution. Some decent progress has already been made in this paradigm of NLP on various tasks (Collobert et al., 2011; Chen and Manning, 2014; Sutskever et al., 2014).

3 Transition-based Dependency Parsing: A Neural Network Architecture

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 aims to predict a transition sequence from an initial parser state to the terminal state. Each state is conventionally regarded as a *configuration*,

[1] In this paper, these two terms are used interchangeably.

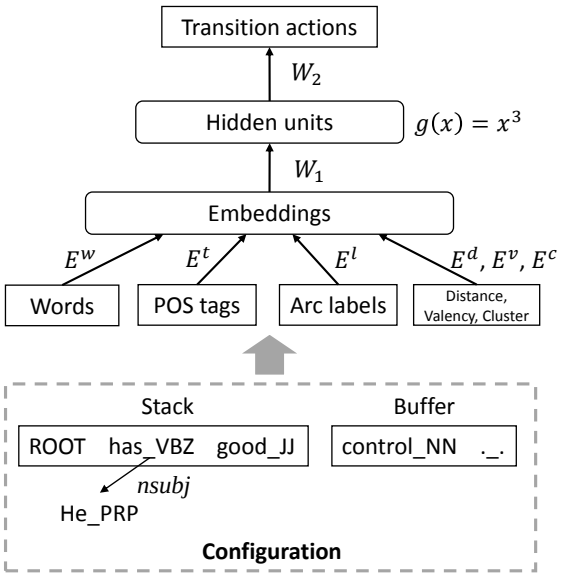

Figure 2: Neural network model for dependency parsing. The *Cluster* features are introduced in Section 5.2.

which typically consists 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 a tuple: $\langle [w_0]_S, [w_1 w_2, ..., w_n]_B, \varnothing \rangle$, 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. We consider the *arc-standard* algorithm (Nivre, 2004) in this paper, which defines three types of transition actions: LEFT-ARC(l), RIGHT-ARC(l), and SHIFT, l is the dependency label.

The typical approach for greedy *arc-standard* parsing is to build a multi-class classifier (e.g., SVM, MaxEnt) 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 and Manning, 2014), the neural network model provides a potential solution.

The architecture of the neural network-based dependency parsing model is illustrated in Figure 2. Primarily, three types of information are extracted from a *configuration* in Chen and Manning's model: word features, POS features and label features respectively. In this study, we add *distance* features indicating the distance between two items, and *valency* features indicating the number of children for a given item (Zhang and Nivre,

Word features
$E^w_{S_i}, E^w_{B_i}, i = 0, 1, 2$
$E^w_{lc1(S_i)}, E^w_{rc1(S_i)}, E^w_{lc2(S_i)}, E^w_{rc2(S_i)}, i = 0, 1$
$E^w_{lc1(lc1(S_i))}, E^w_{rc1(rc1(S_i))}, i = 0, 1$
POS features
$E^t_{S_i}, E^t_{B_i}, i = 0, 1, 2$
$E^t_{lc1(S_i)}, E^t_{rc1(S_i)}, E^t_{lc2(S_i)}, E^t_{rc2(S_i)}, i = 0, 1$
$E^t_{lc1(lc1(S_i))}, E^t_{rc1(rc1(S_i))}, i = 0, 1$
Label features
$E^l_{lc1(S_i)}, E^l_{rc1(S_i)}, E^l_{lc2(S_i)}, E^l_{rc2(S_i)}, i = 0, 1$
$E^l_{lc1(lc1(S_i))}, E^l_{rc1(rc1(S_i))}, i = 0, 1$
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 parsing model. $E^w_p, E^t_p, E^l_p, E^d_p, E^{lv}_p, E^{rv}_p$ indicate the {word, POS, label, distance, left/right valency} embeddings of the element at position p, correspondingly. $lc1$ / $rc1$ is the first child in the left / right, $lc2$ / $rc2$ is the second child in the left / right. S_i and B_i refer to the i^{th} elements respectively in the *stack* and *buffer*.

2011). All of these features are projected to an embedding layer via corresponding 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 a **cube activation function**: $g(x) = x^3$.

The cube activation function can be viewed as a special case of low-rank tensor. Formally, $g(x)$ can be expanded as:

$$g(w_1x_1 + ... + w_mx_m + b) = \sum_{i,j,k}(w_iw_jw_k)x_ix_jx_k + \sum_{i,j} b(w_iw_j)x_ix_j + ...$$

If we treat the bias term as $b \times x_0$ where $x_0 = 1$, then the weight corresponding to each feature combination $x_ix_jx_k$ is $w_iw_jw_k$, which is exactly the same as a rank-1 component tensor in the low-rank form using CP tensor decomposition (Cao and Khudanpur, 2014). Consequently, the cube activation function implicitly derives full feature combinations. An advantage of the cube activation function is that it is flexible for adding extra features to the input. 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.

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, label, distance, valency) embeddings are directly transferable between languages,[2] 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.

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

Various approaches have been studied for learning word embeddings from large-scale plain texts. In this study, we consider the Continuous Bag-of-Words (CBOW) model (Mikolov et al., 2013) as implemented in the open-source toolkit word2vec.[3] 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. 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.

[2]POS tags are language-independent here since we use the universal POS tags (Section 5).

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

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:

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

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*, and then set the averaged vector as the embedding of w_{oov}^T. Formally,

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

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

4.3 Canonical Correlation Analysis

The second approach we consider is similar to Faruqui and Dyer (2014), which use 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.

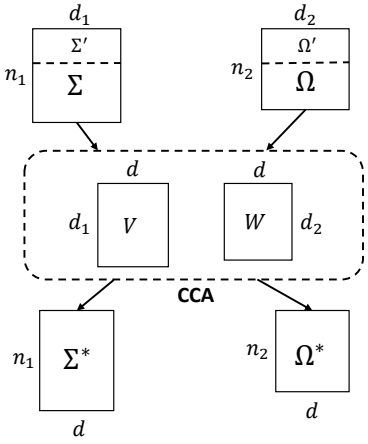

Figure 3: CCA for cross-lingual word representation learning.

Let's treat CCA as a blackbox here, and see how to apply CCA 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 $\mathcal{A}^{S|T}$.[4] 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. Denoting the dimension of resulting word embeddings by $d \le min(d_1, d_2)$. 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:

$$V, W = CCA(\Sigma', \Omega') \quad (3)$$

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

$$\Sigma^* = \Sigma V, \quad \Omega^* = \Omega W \quad (4)$$

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.

Contrary to the 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.

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

Note that both approaches can be generalize 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 (Koehn and Knight, 2002; Mann and Yarowsky, 2001; Haghighi et al., 2008), or from resources like Wiktionary[5] and Panlex.[6]

5 Experiments

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.[7] For French, we combined the WMT-2011 and WMT-2012 monolingual news corpora.[8] We obtained 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}.[9]

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	Dist.	Val.	Cluster
Dim.	50	50	50	5	5	8

Table 2: Dimensions of feature embeddings.

Adaptive stochastic gradient descent (AdaGrad) (Duchi et al., 2011) is used for optimization. For the CCA approach, we use the implementation of Faruqui and Dyer (2014). The dimensions of the monolingual embeddings (d_1, d_2) and the resulting bilingual embeddings are set to 50 equally.

We employ the universal dependency treebanks proposed by McDonald et al. (2013) 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., 2011) which contains 12 POS tags, as well as the universal dependencies which contains 42 relations. We follow the standard split of the treebanks for every language (DE, ES, and FR).[10]

[5] https://www.wiktionary.org/
[6] http://panlex.org/
[7] http://www.statmt.org/wmt11/
[8] http://www.statmt.org/wmt12/
[9] http://www.statmt.org/europarl/
[10] http://code.google.com/p/uni-dep-tb/.

5.2 Baseline Systems

We compare our approach with three systems. For the first baseline, we evaluate the delexicalized transfer of our parser [DELEX], in which we only use non-lexical features.

We also compare our approach with the delexicalized parser in McDonald et al. (2013) [McD13], who used a perceptron-trained transition-based parser with a beam of size 8, along with rich non-local features (Zhang and Nivre, 2011).

Furthermore, we augment cross-lingual word clusters to the perceptron-based delexicalized parser, as proposed in Täckström et al. (2012). We use the same alignment dictionary as described in Section 4 to induce the cross-lingual word clusters. We re-implement the PROJECTED cluster approach in Täckström et al. (2012), which assigns a target word to the cluster with which it is most often aligned:

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

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 the clustering model of Uszkoreit and Brants (2008), we use Brown clustering (Brown et al., 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.

Our delexicalized system obtains slightly lower performance than those reported in McDonald et al. (2013) (McD13), because we're using

Before this dataset was carried out, the CoNLL multilingual dependency treebanks (Buchholz and Marsi, 2006) were often used for evaluation. However, the major problem is that the dependency annotations vary for different languages (e.g. the choice of lexical versus functional head), which makes it impossible to evaluate the LAS.

	Unlabeled Attachment Score (UAS)					Labeled Attachment Score (LAS)				
	EN	DE	ES	FR	AVG	EN	DE	ES	FR	AVG
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). * 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
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 at $p < 0.01$.[12]

greedy decoding and local training. Our re-implementation of (McDonald et al., 2013) 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 UAS and LAS.

Interestingly, we notice that PROJ consistently performs better than CCA by a significant margin, and is comparable to McD13*+Cluster. We will give further analysis to this observation in Section 5.3.1 and 5.3.2.

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 template used here is similar to the POS tag feature templates:

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

Table 5: Word cluster feature templates.

As shown in Table 3, additive improvements are obtained for both PROJ and CCA. 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* augmented with cluster features 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. 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.

5.3.2 Fine-tuning of Word Embeddings

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

		Simple	Robust	Δ
DE	coverage	91.37	94.70	+3.33
	UAS	59.74	60.35	+0.61
	LAS	50.84	51.54	+0.70
ES	coverage	94.51	96.67	+2.16
	UAS	70.97	71.90	+0.93
	LAS	61.34	62.28	+0.94
FR	coverage	90.83	97.60	+6.77
	UAS	71.17	72.93	+1.76
	LAS	61.72	63.12	+1.40

Table 6: Effect of robust projection.

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.

		Fix	Fine-tune	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 et al., 2014) [BIAE], the bilingual compositional vector model (Hermann and Blunsom, 2014) [BICVM], and the bilingual bag-of-words approach (Gouws et al., 2014) [BILBOWA].

For MTL and BIAE, we adopt their released word embeddings directly due to the inefficiency of training.[13] For BICVM and BILBOWA, we rerun their systems on the same dataset as our previous experiments.[14] 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 are optimized using semantic-related objectives. So we suggest that they are probably not well fit for syntactic tasks.

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 Related Studies

Existing approaches for cross-lingual dependency parsing can be divided into three categories: cross-lingual annotation projection methods, jointly modeling methods and cross-lingual representation learning methods.

The cross-lingual annotation projection method is first proposed in Yarowsky et al. (2001) for shallower NLP tasks (POS tagging, NER, etc.). The central idea is to project the syntactic annotations from a resource-rich language to the target language through word alignments, and then train a supervised parser on the projected noisy annotations (Hwa et al., 2005; Smith and Eisner, 2009; Zhao et al., 2009; Jiang et al., 2011; Tiedemann, 2014; Tiedemann, 2015). Noises and errors introduced by the word alignment and annotation projection processes can be reduced with robust projection methods by using graph-based label propagation (Das and Petrov, 2011; Kim and Lee, 2012), or by incorporating auxiliary resources (Kim et al., 2012; Khapra et al., 2010).

The jointly modeling methods integrates the monolingual grammar induction with bilingually-projected dependency information (Liu et al., 2013), or linguistic constraints via posterior

[13] The MTL embeddings are normalized before training.

[14] BICVM only uses the bilingual parallel dataset.

	DE		ES		FR	
	UAS	LAS	UAS	LAS	UAS	LAS
MTL (Klementiev et al., 2012)‡	57.70	47.13	68.04	58.78	67.66	57.30
BIAE (Chandar et al., 2014)‡	53.74	43.68	58.81	46.66	60.10	49.47
BICVM (Hermann and Blunsom, 2014)	56.30	46.99	67.78	58.08	69.13	58.13
BILBOWA (Gouws et al., 2014)	51.65	41.83	65.02	54.35	63.35	51.65
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.

regularization (Ganchev et al., 2009), manually constructed universal dependency parsing rules (Naseem et al., 2010) and manually specified typological features (Naseem et al., 2012). Besides dependency parsing, the joint modeling method has also been applied for other multilingual NLP tasks, including NER (Che et al., 2013; Wang and Manning, 2014), SRL (Zhuang and Zong, 2010; Titov and Klementiev, 2012) and WSD (Guo and Diab, 2010).

The cross-lingual representation learning method aims at building connections across different languages by inducing language-independent feature representations. After that, a parser can be trained at the source-language side within the induced feature space, and directly be applied to the target language. Typical approaches include cross-lingual word clustering (Täckström et al., 2012) which is employed in this paper as a baseline, projection features (Durrett et al., 2012). Xiao and Guo (2014) learns cross-lingual word embeddings and apply them with MSTParser for linguistic transfer, which inspires this work.

It is worth mentioning that remarkable results on the universal dependency treebanks have been achieved by using annotation projection method (Tiedemann, 2014), treebank translation method (Tiedemann and Nivre, 2014), and distribution transferring method (Ma and Xia, 2014). Unlike our approach, all of these methods involve training a parser at the target language side. Parallel bitexts are required in these methods, which limits their scalability to lower-resource languages. That said, these methods have the advantage that they are capable of capturing some language-specific syntactic patterns which our approach cannot.[15] These two kinds of approaches are complementary, and can be integrated to push the performance further.

7 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: 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, a joint method. 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 system on all languages, by an average of 10.9% error reduction on LAS, and further significantly outperforms McDonald et al. (2013) augmented with projected cluster features.[16]

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.

[15] For example, in Spanish and French, adjectives often appears after nouns, thus forming a right-directed arc labeled by *amod*, whereas in English, the *amod* arcs are mostly left-directed.

[16] Our system is publicly available at `https://github.com/jiangfeng1124/acl15-clnndep`.

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.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *CoNLL*, pages 149–164.

Yuan Cao and Sanjeev Khudanpur. 2014. Online learning in tensor space. In *ACL*, pages 666–675.

Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *NIPS*, pages 1853–1861.

Wanxiang Che, Mengqiu Wang, Christopher D. Manning, and Ting Liu. 2013. Named entity recognition with bilingual constraints. In *NAACL*, pages 52–62.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *ACL*, pages 600–609.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*, volume 6, pages 449–454.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic transfer using a bilingual lexicon. In *EMNLP*, pages 1–11, July.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *ACL*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *EACL*, pages 462–471.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *ACL-IJCNLP*, pages 369–377.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2014. Bilbowa: Fast bilingual distributed representations without word alignments. *arXiv preprint arXiv:1410.2455*.

Weiwei Guo and Mona Diab. 2010. Combining orthogonal monolingual and multilingual sources of evidence for all words wsd. In *ACL*, pages 1542–1551, July.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In *EMNLP*, pages 110–120.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *ACL*, volume 2008, pages 771–779.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *ACL*, pages 58–68, June.

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.

Wenbin Jiang, Qun Liu, and Yajuan Lv. 2011. Relaxed cross-lingual projection of constituent syntax. In *EMNLP*, pages 1192–1201.

Mitesh Khapra, Saurabh Sohoney, Anup Kulkarni, and Pushpak Bhattacharyya. 2010. Value for money: Balancing annotation effort, lexicon building and accuracy for multilingual wsd. In *COLING*, pages 555–563.

Seokhwan Kim and Gary Geunbae Lee. 2012. A graph-based cross-lingual projection approach for weakly supervised relation extraction. In *ACL*, pages 48–53.

Sungchul Kim, Kristina Toutanova, and Hwanjo Yu. 2012. Multilingual named entity recognition using parallel data and metadata from wikipedia. In *ACL*, pages 694–702.

Dan Klein and Christopher D Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *ACL*, page 478.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *COLING*, pages 1459–1474.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition-Volume 9*, pages 9–16.

Kai Liu, Yajuan Lü, Wenbin Jiang, and Qun Liu. 2013. Bilingually-guided monolingual dependency grammar induction. In *ACL*, pages 1063–1072.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *ACL*, pages 1337–1348.

Gideon S Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *NAACL*, pages 1–8.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *EMNLP-CoNLL*, pages 122–131.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *EMNLP*, pages 62–72.

Ryan T McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *ACL*, pages 92–97.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *EMNLP*, pages 1234–1244.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *ACL*, pages 629–637.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

David A Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *EMNLP*, pages 822–831. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *NAACL*, pages 477–487.

Jörg Tiedemann and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. *CoNLL-2014*, page 130.

Jörg Tiedemann. 2014. Rediscovering annotation projection for cross-lingual parser induction. In *Proc. COLING*.

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

Ivan Titov and Alexandre Klementiev. 2012. Crosslingual induction of semantic roles. In *ACL*, pages 647–656.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*, pages 384–394.

Jakob Uszkoreit and Thorsten Brants. 2008. Distributed word clustering for large scale class-based language modeling in machine translation. In *ACL*, pages 755–762.

Mengqiu Wang and Christopher D. Manning. 2013. Effect of non-linear deep architecture in sequence labeling. In *IJCNLP*, pages 1285–1291.

Mengqiu Wang and Christopher D Manning. 2014. Cross-lingual projected expectation regularization for weakly supervised learning. *TACL*, 2:55–66.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *CoNLL*, pages 119–129.

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*, pages 1–8.

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

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *ACL*, pages 188–193.

Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. 2009. Cross language dependency parsing using a bilingual lexicon. In *ACL-IJCNLP*, pages 55–63.

Tao Zhuang and Chengqing Zong. 2010. Joint inference for bilingual semantic role labeling. In *EMNLP*, pages 304–314.