Unsupervised Dependency Parsing with Transferring Distribution via Parallel Guidance and Entropy Regularization

Xuezhe Ma
Department of Linguistics
University of Washington
Seattle, WA 98195, USA
xzma@uw.edu

Fei Xia
Department of Linguistics
University of Washington
Seattle, WA 98195, USA
fxia@uw.edu

Abstract

We present a novel approach for inducing unsupervised dependency parsers for languages that have no labeled training data, but have translated text in a resource-rich language. We train probabilistic parsing models for resource-poor languages by transferring cross-lingual knowledge from resource-rich language with entropy regularization. Our method can be used as a purely monolingual dependency parser, requiring no human translations for the test data, thus making it applicable to a wide range of resource-poor languages. We perform experiments on three Data sets — Version 1.0 and version 2.0 of Google Universal Dependency Treebanks and Treebanks from CoNLL shared-tasks, across ten languages. We obtain state-of-the art performance of all the three data sets when compared with previously studied unsupervised and projected parsing systems.

1 Introduction

In recent years, dependency parsing has gained universal interest due to its usefulness in a wide range of applications such as synonym generation (Shinyama et al., 2002), relation extraction (Nguyen et al., 2009) and machine translation (Katz-Brown et al., 2011; Xie et al., 2011). Several supervised dependency parsing algorithms (Nivre and Scholz, 2004; McDonald et al., 2005a; McDonald et al., 2005b; McDonald and Pereira, 2006; Carreras, 2007; Koo and Collins, 2010; Ma and Zhao, 2012; Zhang et al., 2013) have been proposed and achieved high parsing accuracies on several treebanks, due in large part to the availability of dependency treebanks in a number of languages (McDonald et al., 2013). However, the manually annotated treebanks that these parsers rely on are highly expensive to create, in particular when we want to build treebanks for resource-poor languages. This led to a vast amount of research on unsupervised grammar induction (Carroll and Charniak, 1992; Klein and Manning, 2004; Smith and Eisner, 2005; Cohen and Smith, 2009; Spitkovsky et al., 2010; Blunsom and Cohn, 2010; Mareček and Straka, 2013; Spitkovsky et al., 2013), which appears to be a natural solution to this problem, as unsupervised methods require only unannotated text for training parsers. Unfortunately, the unsupervised grammar induction systems' parsing accuracies often significantly fall behind those of supervised systems (McDonald et al., 2011). Furthermore, from a practical standpoint, it is rarely the case that we are completely devoid of resources for most languages.

In this paper, we consider a practically motivated scenario, in which we want to build statistical parsers for resource-poor target languages, using existing resources from a resource-rich source language (like English).[1] We assume that there are absolutely no labeled training data for the target language, but we have access to parallel data with a resource-rich language and a sufficient amount of labeled training data to build an accurate parser for the resource-rich language. This scenario appears similar to the setting in bilingual text parsing. However, most bilingual text parsing approaches require bilingual treebanks — treebanks that have manually annotated tree structures on both sides of source and target languages (Smith and Smith, 2004; Burkett and Klein, 2008), or have tree structures on the source side and translated sentences in the target languages (Huang et

[1] For the sake of simplicity, we refer to the resource-poor language as the "target language", and resource-rich language as the "source language". In addition, in this study we use English as the source resource-rich language, but our methodology can be applied to any resource-rich languages.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 1337–1348,
Baltimore, Maryland, USA, June 23-25 2014.

al., 2009; Chen et al., 2010). Obviously, bilingual treebanks are much more difficult to acquire than the resources required in our scenario, since the labeled training data and the parallel text in our case are completely separated. What is more important is that most studies on bilingual text parsing assumed that the parser is applied only on bilingual text. But our goal is to develop a parser that can be used in completely monolingual setting for each target language of interest.

This scenario is applicable to a large set of languages and many research studies (Hwa et al., 2005) have been made on it. Ganchev et al. (2009) presented a parser projection approach via parallel text using the posterior regularization framework (Graca et al., 2007). McDonald et al. (2011) proposed two parser transfer approaches between two different languages — one is directly transferred parser from delexicalized parsers, and the other parser is transferred using constraint driven learning algorithm where constraints are drawn from parallel corpora. In that work, they demonstrate that even the directly transferred delexicalized parser produces significantly higher accuracies than unsupervised parsers. Cohen et al. (2011) proposed an approach for unsupervised dependency parsing with non-parallel multilingual guidance from one or more helper languages, in which parallel data is not used.

In this work, we propose a learning framework for transferring dependency grammars from a resource-rich language to resource-poor languages via parallel text. We train probabilistic parsing models for resource-poor languages by maximizing a combination of likelihood on parallel data and confidence on unlabeled data. Our work is based on the learning framework used in Smith and Eisner (2007), which is originally designed for parser bootstrapping. We extend this learning framework so that it can be used to transfer cross-lingual knowledge between different languages.

Throughout this paper, English is used as the source language and we evaluate our approach on ten target languages — Danish (da), Dutch (nl), French (fr), German (de), Greek (el), Italian (it), Korean (ko), Portuguese (pt), Spanish (es) and Swedish (sv). Our approach achieves significant improvement over previous state-of-the-art unsupervised and projected parsing systems across all the ten languages, and considerably bridges the gap to fully supervised dependency parsing performance.

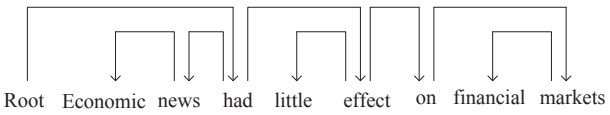

Figure 1: An example dependency tree.

2 Our Approach

Dependency trees represent syntactic relationships through labeled directed edges between heads and their dependents. For example, Figure 1 shows a dependency tree for the sentence, *Economic news had little effect on financial markets*, with the sentence's root-symbol as its root. The focus of this work is on building dependency parsers for target languages, assuming that an accurate English dependency parser and some parallel text between the two languages are available. Central to our approach is a maximizing likelihood learning framework, in which we use an English parser and parallel text to estimate the "transferring distribution" of the target language parsing model (See Section 2.2 for more details). Another advantage of the learning framework is that it combines both the likelihood on parallel data and confidence on unlabeled data, so that both parallel text and unlabeled data can be utilized in our approach.

2.1 Edge-Factored Parsing Model

In this paper, we will use the following notation: $\boldsymbol{x}$ represents a generic input sentence, and $\boldsymbol{y}$ represents a generic dependency tree. $\mathrm{T}(\boldsymbol{x})$ is used to denote the set of possible dependency trees for sentence $\boldsymbol{x}$. The probabilistic model for dependency parsing defines a family of conditional probability $p_\lambda(\boldsymbol{y}|\boldsymbol{x})$ over all $\boldsymbol{y}$ given sentence $\boldsymbol{x}$, with a log-linear form:

$$p_\lambda(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left\{\sum_j \lambda_j F_j(\boldsymbol{y}, \boldsymbol{x})\right\} \quad (1)$$

where F_j are feature functions, $\lambda = (\lambda_1, \lambda_2, \ldots)$ are parameters of the model, and $Z(\boldsymbol{x})$ is a normalization factor, which is commonly referred to as the *partition function*:

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathrm{T}(\boldsymbol{x})} \exp\left\{\sum_j \lambda_j F_j(\boldsymbol{y}, \boldsymbol{x})\right\} \quad (2)$$

A common strategy to make this parsing model efficiently computable is to *factor* dependency trees into sets of edges:

$$F_j(\boldsymbol{y}, \boldsymbol{x}) = \sum_{e \in y} f_j(e, \boldsymbol{x}). \quad (3)$$

That is, dependency tree y is treated as a set of edges e and each feature function $F_j(\boldsymbol{y}, \boldsymbol{x})$ is equal to the sum of all the features $f_j(e, \boldsymbol{x})$.

We denote the *weight function* of each edge e as follows:

$$w(e, \boldsymbol{x}) = \exp\left\{ \sum_j \lambda_j f_j(e, \boldsymbol{x}) \right\} \quad (4)$$

and the conditional probability $p_\lambda(\boldsymbol{y}|\boldsymbol{x})$ has the following form:

$$p_\lambda(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{e \in y} w(e, \boldsymbol{x}) \quad (5)$$

2.2 Model Training

One of the most common model training methods for supervised dependency parser is Maximum conditional likelihood estimation. For a supervised dependency parser with a set of training data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$, the logarithm of the likelihood (a.k.a. the log-likelihood) is given by:

$$L(\lambda) = \sum_i \log p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \quad (6)$$

Maximum likelihood training chooses parameters such that the log-likelihood $L(\lambda)$ is maximized.

However, in our scenario we have no labeled training data for target languages but we have some parallel and unlabeled data plus an English dependency parser. For the purpose of transferring cross-lingual information from the English parser via parallel text, we explore the model training method proposed by Smith and Eisner (2007), which presented a generalization of K function (Abney, 2004), and related it to another semi-supervised learning technique, entropy regularization (Jiao et al., 2006; Mann and McCallum, 2007). The objective K function to be minimized is actually the *expected* negative log-likelihood:

$$\begin{aligned} K &= -\sum_i \sum_{\boldsymbol{y}_i} \tilde{p}(\boldsymbol{y}_i|\boldsymbol{x}_i) \log p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \\ &= \sum_i D(\tilde{p}_i || p_{\lambda,i}) + H(\tilde{p}_i) \end{aligned} \quad (7)$$

where $\tilde{p}_i(\cdot) \stackrel{def}{=} \tilde{p}(\cdot|\boldsymbol{x}_i)$ and $p_{\lambda,i}(\cdot) \stackrel{def}{=} p_\lambda(\cdot|\boldsymbol{x}_i)$. $\tilde{p}(\boldsymbol{y}|\boldsymbol{x})$ is the "transferring distribution" that reflects our uncertainty about the true labels, and we are trying to learn a parametric model $p_\lambda(\boldsymbol{y}|\boldsymbol{x})$ by minimizing the K function.

In our scenario, we have a set of aligned parallel data $P = \{\boldsymbol{x}_i^s, \boldsymbol{x}_i^t, a_i\}$ where a_i is the word alignment for the pair of source-target sentences $(\boldsymbol{x}_i^s, \boldsymbol{x}_i^t)$, and a set of unlabeled sentences of the target language $U = \{\boldsymbol{x}_i^t\}$. We also have a trained English parsing model $p_{\lambda_E}(\boldsymbol{y}|\boldsymbol{x})$. Then the K in equation (7) can be divided into two cases, according to whether $\boldsymbol{x}_i$ belongs to parallel data set P or unlabeled data set U. For the unlabeled examples $\{\boldsymbol{x}_i \in U\}$, some previous studies (e.g., (Abney, 2004)) simply use a uniform distribution over labels (e.g., parses), to reflect that the label is unknown. We follow the method in Smith and Eisner (2007) and take the transferring distribution $\tilde{p}_i$ to be the *actual* current belief $p_{\lambda,i}$. The total contribution of the *unsupervised* examples to K then simplifies to $K_U = \sum_{\boldsymbol{x}_i \in U} H(p_{\lambda,i})$, which may be regarded as the entropy item used to constrain the model's uncertainty H to be low, as presented in the work on entropy regularization (Jiao et al., 2006; Mann and McCallum, 2007).

But how can we define the transferring distribution for the parallel examples $\{\boldsymbol{x}_i^t \in P\}$? We define the transferring distribution by defining the *transferring weight* utilizing the English parsing model $p_{\lambda_E}(\boldsymbol{y}|\boldsymbol{x})$ via parallel data with word alignments:

$$\tilde{w}(e^t, \boldsymbol{x}_i^t) = \begin{cases} w_E(e^s, \boldsymbol{x}_i^s), & \text{if } e^t \stackrel{align}{\longrightarrow} e^s \\ w_E(e_{delex}^t, \boldsymbol{x}_i^s), & \text{otherwise} \end{cases} \quad (8)$$

where $w_E(\cdot,\cdot)$ is the weight function of the English parsing model $p_{\lambda_E}(\boldsymbol{y}|\boldsymbol{x})$, and e_{delex}^t is the delexicalized form[2] of the edge e^t. From the definition of the transferring weight, we can see that, if an edge e^t of the target language sentence $\boldsymbol{x}_i^t$ is aligned to an edge e^s of the English sentence $\boldsymbol{x}_i^s$, we transfer the weight of edge e^t to the corresponding weight of edge e^s in the English parsing model $p_{\lambda_E}(\boldsymbol{y}|\boldsymbol{x})$. If the edge e^t is not aligned to any edges of the English sentence $\boldsymbol{x}_i^s$, we reduce the edge e^t to the delexicalized form and calculate the transferring weight in the English parsing model. There are two advan-

[2]The delexicalized form of an edge is an edge for which only delexicalized features are considered.

tages for this definition of the transferring weight. First, by transferring the weight function to the corresponding weight in the well-developed English parsing model, we can project syntactic information across language boundaries. Second, McDonald et al. (2011) demonstrates that parsers with only delexicalized features produce considerably high parsing performance. By reducing unaligned edges to their delexicalized forms, we can still use those delexicalized features, such as part-of-speech tags, for those unaligned edges, and can address problem that automatically generated word alignments include errors.

From the definition of transferring weight in equation (8), the transferring distribution can be defined in the following way:

$$\tilde{p}(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{\tilde{Z}(\boldsymbol{x})} \prod_{e \in y} \tilde{w}(e, \boldsymbol{x}) \tag{9}$$

where

$$\tilde{Z}(\boldsymbol{x}) = \sum_{y} \prod_{e \in y} \tilde{w}(e, \boldsymbol{x}) \tag{10}$$

Due to the normalizing factor $\tilde{Z}(\boldsymbol{x})$, the transferring distribution is a valid one.

We introduce a multiplier γ as a trade-off between the two contributions (parallel and unsupervised) of the objective function K, and the final objective function $K^{'}$ has the following form:

$$\begin{aligned} K^{'} &= -\sum_{x_i \in P} \sum_{y_i} \tilde{p}(\boldsymbol{y}_i|\boldsymbol{x}_i) \log p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \\ &\quad + \gamma \sum_{x_i \in U} H(p_{\lambda,i}) \\ &= K_P + \gamma K_U \end{aligned} \tag{11}$$

K_P and K_U are the contributions of the parallel and unsupervised data, respectively. One may regard γ as a Lagrange multiplier that is used to constrain the parser's uncertainty H to be low, as presented in several studies on entropy regularization (Brand, 1998; Grandvalet and Bengio, 2004; Jiao et al., 2006).

2.3 Algorithms and Complexity for Model Training

To train our parsing model, we need to find out the parameters λ that minimize the objective function $K^{'}$ in equation (11). This optimization problem is typically solved using quasi-Newton numerical methods such as L-BFGS (Nash and Nocedal, 1991), which requires efficient calculation of the objective function and the gradient of the objective function.

The first item (K_P) of the $K^{'}$ function in equation (11) can be rewritten in the following form:

$$\begin{aligned} K_P &= -\sum_{\boldsymbol{x}_i \in P} \Big[\sum_{y_i} \tilde{p}(\boldsymbol{y}_i|\boldsymbol{x}_i) \sum_{e \in \boldsymbol{y}_i} \log w(e, \boldsymbol{x}_i) \\ &\quad - \log Z(\boldsymbol{x}_i) \Big] \end{aligned} \tag{12}$$

and according to equation (1) and (3) the gradient of K_P can be written as:

$$\begin{aligned} \frac{\partial K_P}{\partial \lambda_j} &= \sum_{x_i \in P} \frac{\partial \tilde{p}(\boldsymbol{y}_i|\boldsymbol{x}_i) \log p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i)}{\partial \lambda_j} \\ &= \sum_{x_i \in P} \Bigg[\sum_{y_i} \tilde{p}(\boldsymbol{y}_i|\boldsymbol{x}_i) \sum_{e \in y_i} f_j(e, \boldsymbol{x}_i) \\ &\quad - \sum_{y_i} p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \sum_{e \in y_i} f_j(e, \boldsymbol{x}_i) \Bigg] \end{aligned} \tag{13}$$

According to equation (9), $\tilde{p}(\boldsymbol{y}|\boldsymbol{x})$ can also be factored into the multiplication of the weight of each edge, so both K_P and its gradient can be calculated by running the $O(n^3)$ inside-outside algorithm (Baker, 1979; Paskin, 2001) for projective parsing. For non-projective parsing, the analogy to the inside algorithm is the $O(n^3)$ matrix-tree algorithm based on Kirchhoff's Matrix-Tree Theorem, which is dominated asymptotically by a matrix determinant (Koo et al., 2007; Smith and Smith, 2007). The gradient of a determinant may be computed by matrix inversion, so evaluating the gradient again has the same $O(n^3)$ complexity as evaluating the function.

The second item (K_U) of the $K^{'}$ function in equation (11) is the Shannon entropy of the posterior distribution over parsing trees, and can be written into the following form:

$$\begin{aligned} K_U &= -\sum_{\boldsymbol{x}_i \in U} \Big[\sum_{y_i} p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \sum_{e \in \boldsymbol{y}_i} \log w(e, \boldsymbol{x}_i) \\ &\quad - \log Z(\boldsymbol{x}_i) \Big] \end{aligned} \tag{14}$$

and the gradient of K_U is in the following:

$$\begin{aligned} \frac{\partial K_U}{\partial \lambda_j} &= \sum_{x_i \in U} \frac{\partial p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \log p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i)}{\partial \lambda_j} \\ &= -\sum_{y_i} p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \log p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) F_j(\boldsymbol{y}_i, \boldsymbol{x}_i) \\ &\quad + \Big(\sum_{y_i} p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \log p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) \Big) \\ &\quad \cdot \Big(\sum_{y_i} p_\lambda(\boldsymbol{y}_i|\boldsymbol{x}_i) F_j(\boldsymbol{y}_i, \boldsymbol{x}_i) \Big) \end{aligned} \tag{15}$$

	#sents/#tokens		
	training	dev	test
	Version 1.0		
de	2,200/30,460	800/12,215	1,000/16,339
es	3,345/94,232	370/10,191	300/8,295
fr	3,312/74,979	366/8,071	300/6,950
ko	5,308/62,378	588/6,545	298/2,917
sv	4,447/66,631	493/9,312	1,219/20,376
	Version 2.0		
de	14,118/26,4906	800/12,215	1,000/16,339
es	14,138/37,5180	1,569/40,950	300/8,295
fr	14,511/35,1233	1,611/38,328	300/6,950
it	6,389/14,9145	400/9,541	400/9,187
ko	5437/60,621	603/6,438	299/2,631
pt	9,600/23,9012	1,200/29,873	1,198/29,438
sv	4,447/66,631	493/9,312	1,219/20,376

Table 1: Data statistics of two versions of Google Universal Treebanks for the target languages.

Similar with the calculation of K_P, K_U can also be computed by running the inside-outside algorithm (Baker, 1979; Paskin, 2001) for projective parsing. For the gradient of K_U, both the two multipliers of the second item in equation (15) can be computed using the same inside-outside algorithm. For the first item in equation (15), an $O(n^3)$ dynamic programming algorithm that is closely related to the forward-backward algorithm (Mann and McCallum, 2007) for the entropy regularized CRF (Jiao et al., 2006) can be used for projective parsing. For non-projective parsing, however, the runtime rises to $O(n^4)$. In this paper, we focus on projective parsing.

2.4 Summary of Our Approach

To summarize the description in the previous sections, our approach is performed in the following steps:

1. Train an English parsing model $p_{\lambda_E}(\boldsymbol{y}|\boldsymbol{x})$, which is used to estimate the transferring distribution $\tilde{p}(\boldsymbol{y}|\boldsymbol{x})$.

2. Prepare parallel text by running word alignment method to obtain word alignments,[3] and prepare the unlabeled data.

3. Train a parsing model for the target language by minimizing the objective K' function which is the combination of expected negative log-likelihood on parallel and unlabeled data.

[3]The word alignment methods do not require additional resources besides parallel text.

	# sents					
	500	1000	2000	5000	10000	20000
da	12,568	25,225	49,889	126,623	254,565	509,480
de	13,548	26,663	53,170	133,596	265,589	527,407
el	14,198	28,302	56,744	143,753	286,126	572,777
es	15,147	29,214	57,526	144,621	290,517	579,164
fr	15,046	29,982	60,569	153,874	306,332	609,541
it	15,151	29,786	57,696	145,717	288,337	573,557
ko	3,814	7,679	15,337	38,535	77,388	155,051
nl	13,234	26,777	54,570	137,277	274,692	551,463
pt	14,346	28,109	55,998	143,221	285,590	571,109
sv	12,242	24,897	50,047	123,069	246,619	490,086

Table 2: The number of tokens in parallel data used in our experiments. For all these corpora, the other language is English.

3 Data and Tools

In this section, we illustrate the data sets used in our experiments and the tools for data preparation.

3.1 Choosing Target Languages

Our experiments rely on two kinds of data sets: (i) Monolingual Treebanks with consistent annotation schema — English treebank is used to train the English parsing model, and the Treebanks for target languages are used to evaluate the parsing performance of our approach. (ii) Large amounts of parallel text with English on one side. We select target languages based on the availability of these resources. The monolingual treebanks in our experiments are from the Google Universal Dependency Treebanks (McDonald et al., 2013), for the reason that the treebanks of different languages in Google Universal Dependency Treebanks have consistent syntactic representations.

The parallel data come from the Europarl corpus version 7 (Koehn, 2005) and Kaist Corpus[4]. Taking the intersection of languages in the two kinds of resources yields the following seven languages: French, German, Italian, Korean, Portuguese, Spanish and Swedish.

The treebanks from CoNLL shared-tasks on dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007) appear to be another reasonable choice. However, previous studies (McDonald et al., 2011; McDonald et al., 2013) have demonstrated that a homogeneous representation is critical for multilingual language technologies that require consistent cross-lingual analysis for downstream components, and the heterogenous representations used in CoNLL shared-tasks treebanks weaken any conclusion that can be drawn.

[4]`http://semanticweb.kaist.ac.kr/home/index.php/Corpus10`

	DTP	DTP†	PTP†	-U	+U	OR
de	58.50	58.46	69.21	73.72	74.01	78.64
es	68.07	68.72	72.57	75.32	75.60	82.56
fr	70.14	71.13	74.60	76.65	76.93	83.69
ko	42.37	43.57	53.72	59.72	59.94	89.85
sv	70.56	70.59	75.87	78.91	79.27	85.59
Ave	61.93	62.49	69.19	72.86	73.15	84.67

Table 3: UAS for two versions of our approach, together with baseline and oracle systems on Google Universal Treebanks version 1.0. "Ave" is the macro-average across the five languages.

For comparison with previous studies, nevertheless, we also run experiments on CoNLL treebanks (see Section 4.4 for more details). We evaluate our approach on three target languages from CoNLL shared task treebanks, which do not appear in Google Universal Treebanks. The three languages are Danish, Dutch and Greek. So totally we have ten target languages. The parallel data for these three languages are also from the Europarl corpus version 7.

3.2 Word Alignments

In our approach, word alignments for the parallel text are required. We perform word alignments with the open source GIZA++ toolkit[5]. The parallel corpus was preprocessed in standard ways, selecting sentences with the length in the range from 3 to 100. Then we run GIZA++ with the default setting to generate word alignments in both directions. We then make the intersection of the word alignments of two directions to generate one-to-one alignments.

3.3 Part-of-Speech Tagging

Several features in our parsing model involve part-of-speech (POS) tags of the input sentences. The set of POS tags needs to be consistent across languages and treebanks. For this reason we use the universal POS tag set of Petrov et al. (2011). This set consists of the following 12 coarse-grained tags: NOUN (nouns), VERB (verbs), ADJ (adjectives), ADV (adverbs), PRON (pronouns), DET (determiners), ADP (prepositions or postpositions), NUM (numerals), CONJ (conjunctions), PRT (particles), PUNC (punctuation marks) and X (a catch-all for other categories such as abbreviations or foreign words).

POS tags are not available for parallel data in the Europarl and Kaist corpus, so we need to provide the POS tags for these data. In our experiments, we train a Stanford POS Tagger (Toutanova et al., 2003) for each language. The labeled training data for each POS tagger are extracted from the training portion of each Treebanks. The average tagging accuracy is around 95%.

Undoubtedly, we are primarily interested in applying our approach to build statistical parsers for resource-poor target languages without any knowledge. For the purpose of evaluation of our approach and comparison with previous work, we need to exploit the gold POS tags to train the POS taggers. As part-of-speech tags are also a form of syntactic analysis, this assumption weakens the applicability of our approach. Fortunately, some recently proposed POS taggers, such as the POS tagger of Das and Petrov (2011), rely only on labeled training data for English and the same kind of parallel text in our approach. In practice we can use this kind of POS taggers to predict POS tags, whose tagging accuracy is around 85%.

[5] https://code.google.com/p/giza-pp/

	DTP†	PTP†	-U	+U	OR
de	58.56	69.77	73.92	74.30	81.65
es	68.72	73.22	75.21	75.53	83.92
fr	71.13	74.75	76.14	76.53	83.51
it	70.74	76.08	77.55	77.74	85.47
ko	38.55	43.34	59.71	59.89	90.42
pt	69.82	74.59	76.30	76.65	85.67
sv	70.59	75.87	78.91	79.27	85.59
Ave	64.02	69.66	73.96	74.27	85.18

Table 4: UAS for two versions of our approach, together with baseline and oracle systems on Google Universal Treebanks version 2.0. "Ave" is the macro-average across the seven languages.

4 Experiments

In this section, we will describe the details of our experiments and compare our results with previous methods.

4.1 Data Sets

As presented in Section 3.1, we evaluate our parsing approach on both version 1.0 and version 2.0 of Google Univereal Treebanks for seven languages[6]. We use the standard splits of the treebank for each language as specified in the release of the data[7]. Table 1 presents the statistics of the two versions of Google Universal Treebanks. We strip all

[6] Japanese and Indonesia are excluded as no practicable parallel data are available.

[7] https://code.google.com/p/uni-dep-tb/

	Google Universal Treebanks V1.0														
	de			**es**			**fr**			**ko**			**sv**		
# sents	PTP†	-U	+U	PTP†	-U	+U	PTP†	-U	+U	PTP†	-U	+U	PTP†	-U	+U
500	63.23	70.79	70.93	70.09	72.32	72.64	72.24	74.64	74.90	47.71	56.87	57.22	71.70	75.88	76.13
1000	65.61	71.71	71.86	70.90	73.44	73.67	72.95	75.07	75.35	47.83	57.65	58.15	72.38	76.55	77.03
2000	66.52	72.33	72.48	72.01	73.57	73.81	73.69	75.88	76.22	48.37	58.19	58.44	73.65	77.86	78.12
5000	67.79	73.06	73.31	72.34	74.30	74.79	74.31	76.02	76.29	53.02	58.57	59.04	74.88	78.48	78.70
10000	68.44	73.59	73.92	72.48	74.86	75.26	74.43	76.14	76.34	53.61	59.17	59.55	75.34	78.78	79.08
20000	69.21	73.72	74.01	72.57	75.32	75.60	74.60	76.55	76.93	53.72	59.72	59.94	75.87	78.91	79.27

	Google Universal Treebanks V2.0														
	de			**es**			**fr**			**ko**			**it**		
# sents	PTP†	-U	+U	PTP†	-U	+U	PTP†	-U	+U	PTP†	-U	+U	PTP†	-U	+U
500	60.10	71.07	71.39	69.52	72.97	73.28	71.10	74.57	74.70	40.09	56.60	57.10	72.80	75.67	75.94
1000	61.76	72.15	72.39	70.78	73.48	73.79	72.14	75.13	75.43	40.44	57.55	57.93	73.55	76.43	76.67
2000	65.35	72.73	73.04	71.75	74.10	74.35	73.21	75.78	76.06	40.87	58.11	58.43	74.44	76.99	77.39
5000	67.86	73.32	73.62	72.43	74.55	74.83	74.14	75.83	76.02	40.90	58.48	58.96	75.07	77.10	77.34
10000	68.70	73.71	74.02	72.85	74.80	74.95	74.53	75.97	76.17	41.29	59.13	59.44	75.65	77.50	77.71
20000	69.77	73.92	74.30	73.22	75.21	75.53	74.75	76.14	76.53	43.34	59.71	59.89	76.08	77.55	77.74

	pt		
# sents	PTP†	-U	+U
500	71.34	74.41	74.68
1000	71.91	74.48	75.08
2000	72.93	75.10	75.32
5000	73.78	75.88	75.98
10000	74.40	75.99	76.15
20000	74.59	76.30	76.65

Table 5: Parsing results of our approach with different amount of parallel data on Google Universal Treebanks version 1.0 and 2.0. We omit the results of Swedish for treebanks version 2.0 since the data for Swedish from version 2.0 are exactly the same with those from version 1.0.

the dependency annotations off the training portion of each treebank, and use that as the unlabeled data for that target language. We train our parsing model with different numbers of parallel sentences to analyze the influence of the amount of parallel data on the parsing performance of our approach. The parallel data sets contain 500, 1000, 2000, 5000, 10000 and 20000 parallel sentences, respectively. We randomly extract parallel sentences from each corpora, and smaller data sets are subsets of larger ones. Table 2 shows the number of tokens in the parallel data used in the experiments.

4.2 System performance and comparison on Google Universal Treebanks

For the comparison of parsing performance, we run experiments on the following systems:

DTP: The direct transfer parser (DTP) proposed by McDonald et al. (2011), who train a delexicalized parser on English labeled training data with no lexical features, then apply this parser to parse target languages directly. It is based on the transition-based dependency parsing paradigm (Nivre, 2008). We directly cite the results reported in McDonald et al. (2013). In addition to their original results, we also report results by reimplementing the direct transfer parser based on the first-order projective dependency parsing model (McDonald et al., 2005a) (DTP†).

PTP The projected transfer parser (PTP) described in McDonald et al. (2011). The results of the projected transfer parser reimplemented by us is marked as "PTP†".

-U: Our approach training on only parallel data without unlabeled data for the target language. The parallel data set for each language contains 20,000 sentences.

+U: Our approach training on both parallel and unlabeled data. The parallel data sets are the ones contains 20,000 sentences.

OR: the supervised first-order projective dependency parsing model (McDonald et al., 2005a), trained on the original treebanks with maximum likelihood estimation (equation 6). One may regard this system as an oracle of transfer parsing.

Parsing accuracy is measured with unlabeled attachment score (UAS): the percentage of words with the correct head.

Table 3 and Table 4 shows the parsing results of our approach, together with the results of the baseline systems and the oracle, on version 1.0 and version 2.0 of Google Universal Treebanks, respectively. Our approaches significantly outperform all the baseline systems across all the seven target languages. For the results on Google Universal Treebanks version 1.0, the improvement on average over the projected transfer paper (PTP†) is 3.96%

and up to 6.22% for Korean and 4.80% for German. For the other three languages, the improvements are remarkable, too — 2.33% for French, 3.03% for Spanish and 3.40% for Swedish. By adding entropy regularization from unlabeled data, our full model achieves average improvement of 0.29% over the "-U" setting. Moreover, our approach considerably bridges the gap to fully supervised dependency parsers, whose average UAS is 84.67%. For the results on treebanks version 2.0, we can get similar observation and draw the same conclusion.

4.3 Effect of the Amount of Parallel Text

Table 5 illustrates the UAS of our approach trained on different amounts of parallel data, together with the results of the projected transfer parser re-implemented by us (PTP†). We run two versions of our approach for each of the parallel data sets, one with unlabeled data (+U) and the other without them (-U). From table 5 we can get three observations. First, even the parsers trained with only 500 parallel sentences achieve considerably high parsing accuracies (average 70.10% for version 1.0 and 71.59% for version 2.0). This demonstrates that our approach does not rely on a large amount of parallel data. Second, when gradually increasing the amount of parallel data, the parsing performance continues improving. Third, entropy regularization with unlabeled data makes modest improvement on parsing performance over the parsers without unlabeled data. This proves the effectiveness of the entropy regularization from unlabeled data.

4.4 Experiments on CoNLL Treebanks

To make a thorough empirical comparison with previous studies, we also evaluate our system without unlabeled data (-U) on treebanks from CoNLL shared task on dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007). To facilitate comparison, we use the same eight Indo-European languages as target languages: Danish, Dutch, German, Greek, Italian, Portuguese, Spanish and Swedish, and same experimental setup as McDonald et al. (2011). We report both the results of the direct transfer and projected transfer parsers directly cited from McDonald et al. (2011) (DTP and PTP) and re-implemented by us (DTP†and PTP†).

Table 6 gives the results comparing the model without unlabeled data (-U) presented in this work to those five baseline systems and the oracle (OR). The results of unsupervised DMV model (Klein and Manning, 2004) are from Table 1 of McDonald et al. (2011). Our approach outperforms all these baseline systems and achieves state-of-the-art performance on all the eight languages.

	DMV	DTP	DTP†	PTP	PTP†	-U	OR
da	33.4	45.9	46.8	48.2	50.0	50.1	87.1
de	18.0	47.2	46.0	50.9	52.4	57.3	87.0
el	39.9	63.9	62.9	66.8	65.3	67.4	82.3
es	28.5	53.3	54.4	55.8	59.9	60.3	83.6
it	43.1	57.7	59.9	60.8	63.4	64.0	83.9
nl	38.5	60.8	60.7	67.8	66.5	68.2	78.2
pt	20.1	69.2	71.1	71.3	74.8	75.1	87.2
sv	44.0	58.3	60.3	61.3	62.8	66.7	88.0
Ave	33.2	57.0	57.8	60.4	61.9	63.6	84.7

Table 6: Parsing results on treebanks from CoNLL shared tasks for eight target languages. The results of unsupervised DMV model are from Table 1 of McDonald et al. (2011).

In order to compare with more previous methods, we also report parsing performance on sentences of length 10 or less after punctuation has been removed. Table 7 shows the results of our system and the results of baseline systems. "USR†" is the weakly supervised system of Naseem et al. (2010). "PGI" is the phylogenetic grammar induction model of Berg-Kirkpatrick and Klein (2010). Both the results of the two systems are cited from Table 4 of McDonald et al. (2011). We also include the results of the unsupervised dependency parsing model with non-parallel multilingual guidance (NMG) proposed by Cohen et al. (2011)[8], and "PR" which is the posterior regularization approach presented in Gillenwater et al. (2010). All the results are shown in Table 7.

From Table 7, we can see that among the eight target languages, our approach achieves best parsing performance on six languages — Danish, German, Greek, Italian, Portuguese and Swedish. It should be noted that the "NMG" system utilizes more than one helper languages. So it is not directly comparable to our work.

4.5 Extensions

In this section, we briefly outline a few extensions to our approach that we want to explore in future work.

[8]For each language, we use the best result of the four systems in Table 3 of Cohen et al. (2011)

	DTP	DTP†	PTP	PTP†	USR†	PGI	PR	NMG	-U
da	53.2	55.3	57.4	59.8	55.1	41.6	44.0	59.9	**60.1**
de	65.9	57.9	67.0	63.5	60.0	—	—	—	**67.5**
el	73.9	70.8	73.9	72.3	60.3	—	—	73.0	**74.3**
es	58.0	62.3	62.3	66.1	68.3	58.4	62.4	**76.7**	64.6
it	65.5	66.9	69.9	71.5	47.9	—	—	—	**73.6**
nl	67.6	66.0	**72.2**	72.1	44.0	45.1	37.9	50.7	70.5
pt	77.9	79.2	80.6	82.9	70.9	63.0	47.8	79.8	**83.3**
sv	70.4	70.2	71.3	70.4	52.6	58.3	42.2	74.0	**75.1**
Ave	66.6	66.1	69.4	69.8	57.4	—	—	—	**71.1**

Table 7: UAS on sentences of length 10 or less without punctuation from CoNLL shared task treebanks. "USR†" is the weakly supervised system of Naseem et al. (2010). "PGI" is the phylogenetic grammar induction model of Berg-Kirkpatrick and Klein (2010). Both the "USR†" and "PGI" systems are implemented and reported by McDonald et al. (2011). "NMG" is the unsupervised dependency parsing model with non-parallel multilingual guidance (Cohen et al., 2011). "PR" is the posterior regularization approach presented in Gillenwater et al. (2010). Some systems' results for certain target languages are not available as marked by —.

4.5.1 Non-Projective Parsing

As mentioned in section 2.3, the runtime to compute K_U and its gradient is $O(n^4)$. One reasonable speedup, as presented in Smith and Eisner (2007), is to replace Shannon entropy with Rényi entropy. The **Rényi entropy** is parameterized by α:

$$R_\alpha(p) = \frac{1}{1-\alpha} \log \Big(\sum_y p(y)^\alpha \Big) \qquad (16)$$

With Rényi entropy, the computation of K_U and its gradient is $O(n^3)$, even for non-projective case.

4.5.2 Higher-Order Models for Projective Parsing

Our learning framework can be extended to higher-order dependency parsing models. For example, if we want to make our model capable of utilizing more contextual information, we can extend our transferring weight to higher-order parts:

$$\tilde{w}(p^t, \boldsymbol{x}_i^t) = \begin{cases} w_E(p^s, \boldsymbol{x}_i^s), & \text{if } p^t \overset{align}{\longrightarrow} p^s \\ w_E(p^t_{delex}, \boldsymbol{x}_i^s), & \text{otherwise} \end{cases} \qquad (17)$$

where p is a small *part* of tree $\boldsymbol{y}$ that has limited interactions. For projective parsing, several algorithms (McDonald and Pereira, 2006; Carreras, 2007; Koo and Collins, 2010; Ma and Zhao, 2012) have been proposed to solve the model training problems (calculation of objective function and gradient) for different factorizations.

4.5.3 IGT Data

One possible direction to improve our approach is to replace parallel text with Interlinear Glossed Text (IGT) (Lewis and Xia, 2010), which is a semi-structured data type encoding more syntactic information than parallel data. By using IGT Data, not only can we obtain more accurate word alignments, but also extract useful cross-lingual information for the resource-poor language.

5 Conclusion

In this paper, we propose an unsupervised projective dependency parsing approach for resource-poor languages, using existing resources from a resource-rich source language. By presenting a model training framework, our approach can utilize parallel text to estimate transferring distribution with the help of a well-developed resource-rich language dependency parser, and use unlabeled data as entropy regularization. The experimental results on three data sets across ten target languages show that our approach achieves significant improvement over previous studies.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. BCS-0748919. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Steven Abney. 2004. Understanding the Yarowsky algorithm. *Computational Linguistics*, 30:2004.

James K. Baker. 1979. Trainable grammars for speech recognition. In *Proceedings of 97th meeting of the Acoustical Society of America*, pages 547–550.

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic grammar induction. In *Proceedings of ACL-2010*, pages 1288–1297, Uppsala, Sweden, July.

Phil Blunsom and Trevor Cohn. 2010. Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of EMNLP-2010*, pages 1204–1213, Cambridge, MA, October.

Matthew Brand. 1998. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5):1155–1182.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceeding of CoNLL-2006*, pages 149–164, New York, NY.

David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proceedings of EMNLP-2008*, pages 877–886, Honolulu, Hawaii, October.

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CONLL*, pages 957–961.

Glenn Carroll and Eugene Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. In *Proceedings of Working Notes of the Workshop Statistically-Based NLP Techniques*.

Wenliang Chen, Jun'ichi Kazama, and Kentaro Torisawa. 2010. Bitext dependency parsing with bilingual subtree constraints. In *Proceedings of ACL-2010*, pages 21–29, Uppsala, Sweden, July.

Shay Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of NAACL/HLT-2009*, pages 74–82, Boulder, Colorado, June.

Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of EMNLP-2011*, pages 50–61, Edinburgh, Scotland, UK., July.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL/HLT-2011*, pages 600–609, Portland, Oregon, USA, June.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *Proceedings of ACL/AFNLP-2009*, pages 369–377, Suntec, Singapore, August.

Jennifer Gillenwater, Kuzman Ganchev, João Graça, Fernando Pereira, and Ben Taskar. 2010. Sparsity in dependency grammar induction. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 194–199, Uppsala, Sweden, July.

Joao V. Graca, Lf Inesc-id, Kuzman Ganchev, and Ben Taskar. 2007. Expectation maximization and posterior constraints. In *Advances in NIPS*, pages 569–576.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*.

Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of EMNLP-2009*, pages 1222–1231, Singapore, August.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:11–311.

Feng Jiao, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of COLING/ACL-2006*, pages 209–216, Sydney, Australia, July.

Jason Katz-Brown, Slav Petrov, Ryan McDonald, Franz Och, David Talbot, Hiroshi Ichikawa, Masakazu Seno, and Hideto Kazawa. 2011. Training a parser for machine translation reordering. In *Proceedings of EMNLP-2011*, pages 183–192, Edinburgh, Scotland, UK., July.

Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of ACL-2004*, pages 478–485, Barcelona, Spain, July.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of ACL-2010*, pages 1–11, Uppsala, Sweden, July.

Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured predicition models via the matrix-tree theorem. In *Proceedings of EMNLP-CONLL 2007*, pages 141–150, Prague, Czech, June.

William D. Lewis and Fei Xia. 2010. Developing odin: A multilingual repository of annotated language data for hundreds of the world's languages. *LLC*, 25(3):303–319.

Xuezhe Ma and Hai Zhao. 2012. Fourth-order dependency parsing. In *Proceedings of COLING 2012: Posters*, pages 785–796, Mumbai, India, December.

Gideon S. Mann and Andrew McCallum. 2007. Efficient computation of entropy gradient for semi-supervised conditional random fields. In *Proceedings of NAACL/HLT-2007*, pages 109–112, Stroudsburg, PA, USA.

David Mareček and Milan Straka. 2013. Stop-probability estimates computed on a large corpus improve unsupervised dependency parsing. In *Proceedings of ACL-2013*, pages 281–290, Sofia, Bulgaria, August.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL-2006*, pages 81–88, Trento, Italy, April.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of ACL-2005*, pages 91–98, Ann Arbor, Michigan, USA, June 25-30.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT/EMNLP-2005*, pages 523–530, Vancouver, Canada, October.

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

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of ACL-2013*, pages 92–97, Sofia, Bulgaria, August.

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

Stephen G. Nash and Jorge Nocedal. 1991. A numerical study of the limited memory bfgs method and truncated-newton method for large scale optimization. *SIAM Journal on Optimization*, 1(2):358–372.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of EMNLP-2009*, pages 1378–1387, Singapore, August.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of COLING-2004*, pages 64–70, Geneva, Switzerland, August 23-27.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan Mcdonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceeding of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Comput. Linguist.*, 34(4):513–553, December.

Mark A. Paskin. 2001. Cubic-time parsing and learning algorithms for grammatical bigram models. Technical Report, UCB/CSD-01-1148.

Slav Petrov, Dipanjan Das, and Ryan T. McDonald. 2011. A universal part-of-speech tagset. *CoRR*, abs/1104.2086.

Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. 2002. Automatic paraphrase acquisition from news articles. In *Proceeding of HLT-2002*, pages 313–318.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of ACL-2005*, pages 354–362, Ann Arbor, Michigan, June.

David A. Smith and Jason Eisner. 2007. Bootstrapping feature-rich dependency parsers with entropic priors. In *Proceedings of EMNLP/CoNLL-2007*, pages 667–677, Prague, Czech Republic, June.

David A. Smith and Noah A. Smith. 2004. Bilingual parsing with factored estimation: Using English to parse Korean. In *Proceedings of EMNLP-2004*, pages 49–56.

David A. Smith and Noah A. Smith. 2007. Probabilistic models of nonporjective dependency trees. In *Proceedings of EMNLP-CONLL 2007*, pages 132–140, Prague, Czech, June.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Proceedings of NAACL/HLT-2010*, pages 751–759, Los Angeles, California, June.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2013. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In *Proceedings of EMNLP-2013*, pages 1983–1995, Seattle, Washington, USA, October.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL/HLT-2003*, pages 252–259.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proceedings of EMNLP-2011*, pages 216–226, Edinburgh, Scotland, UK., July.

Hao Zhang, Liang Huang, Kai Zhao, and Ryan McDonald. 2013. Online learning for inexact hypergraph search. In *Proceedings of EMNLP-2013*, pages 908–913, Seattle, Washington, USA, October.