# Soft Cross-lingual Syntax Projection for Dependency Parsing

**Zhenghua Li , Min Zhang[*], Wenliang Chen**
Provincial Key Laboratory for Computer Information Processing Technology
Soochow University
{zhli13,minzhang,wlchen}@suda.edu.cn

## Abstract

This paper proposes a simple yet effective framework of soft cross-lingual syntax projection to transfer syntactic structures from source language to target language using monolingual treebanks and large-scale bilingual parallel text. Here, *soft* means that we only project reliable dependencies to compose high-quality target structures. The projected instances are then used as additional training data to improve the performance of supervised parsers. The major issues for this idea are 1) errors from the source-language parser and unsupervised word aligner; 2) intrinsic syntactic non-isomorphism between languages; 3) incomplete parse trees after projection. To handle the first two issues, we propose to use a probabilistic dependency parser trained on the target-language treebank, and prune out unlikely projected dependencies that have low marginal probabilities. To make use of the incomplete projected syntactic structures, we adopt a new learning technique based on *ambiguous labelings*. For a word that has no head words after projection, we enrich the projected structure with all other words as its candidate heads as long as the newly-added dependency does not cross any projected dependencies. In this way, the syntactic structure of a sentence becomes a parse forest (ambiguous labels) instead of a single parse tree. During training, the objective is to maximize the mixed likelihood of manually labeled instances and projected instances with ambiguous labelings. Experimental results on benchmark data show that our method significantly outperforms a strong baseline supervised parser and previous syntax projection methods.

## 1 Introduction

During the past decade, supervised dependency parsing has made great progress. However, due to the limitation of scale and genre coverage of labeled data, it is very difficult to further improve the performance of supervised parsers. On the other hand, it is very time-consuming and labor-intensive to manually construct treebanks. Therefore, lots of recent work has been devoted to get help from bilingual constraints. The motivation behind are two-fold. First, a difficult syntactic ambiguity in one language may be very easy to resolve in another language. Second, a more accurate parser on one language may help an inferior parser on another language, where the performance difference may be due to the intrinsic complexity of languages or the scale of accessible labeled resources.

Following the above research line, much effort has been done recently to explore bilingual constraints for parsing. Burkett and Klein (2008) propose a reranking based method for joint constituent parsing of bitext, which can make use of structural correspondence features in both languages. Their method needs bilingual treebanks with manually labeled syntactic trees on both sides for training. Huang et al. (2009) compose useful parsing features based on word reordering information in source-language sentences. Chen et al. (2010a) derive bilingual subtree constraints with auto-parsed source-language sentences. During training, both Huang et al. (2009) and Chen et al. (2010a) require bilingual text with target-language gold-standard dependency trees. All above work shows significant performance gain

[*] Correspondence author

*Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*,
pages 783–793, Dublin, Ireland, August 23-29 2014.

over monolingual counterparts. However, one potential disadvantage is that bilingual treebanks and bitext with one-side annotation are difficult to obtain. Therefore, They usually conduct experiments on treebanks with a few thousand sentences. To break this constraint, Chen et al. (2011) extend their work in Chen et al. (2010a) and translate text of monolingual treebanks to obtain bilingual treebanks with a statistical machine translation system.

This paper explores another line of research and aims to boost the state-of-the-art parsing accuracy via syntax projection. Syntax projection typically works as follows. First, we train a parser on source-language treebank, called a source parser. Then, we use the source parser to produce automatic syntactic structures on the source side of bitext. Next, with the help of automatic word alignments, we project the source-side syntactic structures into the target side. Finally, the target-side structures are used as gold-standard to train new parsing models of target language. Previous work on syntax projection mostly focuses on unsupervised grammar induction where no labeled data exists for target language (Hwa et al., 2005; Spreyer and Kuhn, 2009; Ganchev et al., 2009; Liu et al., 2013). Smith and Eisner (2009) propose quasi-synchronous grammar for cross-lingual parser projection and assume the existence of hundreds of target language annotated sentences. Similar to our work in this paper, Jiang et al. (2010) try to explore projected structures to further improve the performance of statistical parsers trained on full-scale monolingual treebanks (see Section 4.4 for performance comparison).

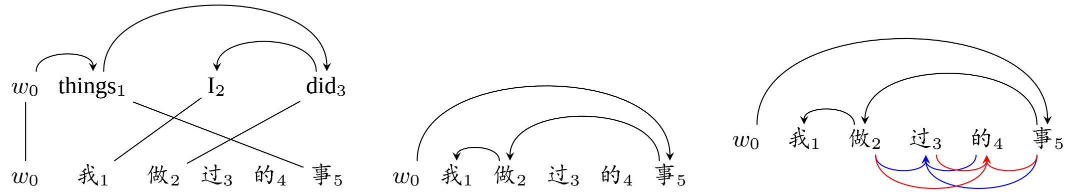
The major issues for syntax projection are 1) errors from the source-language parser and unsupervised word aligner; 2) intrinsic syntactic non-isomorphism between languages; 3) incomplete parse trees after projection. Hwa et al. (2005) propose a simple projection algorithm based on the *direct correspondence assumption* (DCA). They apply post-editing to the projected structures with a set of hand-crafted heuristic rules, in order to handle some typical cross-lingual syntactic divergences. Similarly, Ganchev et al. (2009) manually design several language-specific constrains during projection, and use projected partial structures as soft supervision during training based on posterior regularization (Ganchev et al., 2010). To make use of projected instances with incomplete trees, Spreyer and Kuhn (2009) propose a heuristic method to adapt training procedures of dependency parsing. Instead of directly using incomplete trees to train dependency parsers, Jiang et al. (2010) train a local dependency/non-dependency classifier on projected syntactic structures, and use outputs of the classifier as auxiliary features to help supervised parsers. One potential common drawback of above work is the lack of a systematic way to handle projection errors and incomplete trees.

Different from previous work, this paper proposes a simple yet effective framework of soft syntax projection for dependency parsing, and provides a more elegant and systematic way to handle the above issues. First, we propose to use a probabilistic parser trained on target-language treebank, and prune unlikely projected dependencies which have very low marginal probabilities. Second, we adopt a new learning technique based on ambiguous labelings to make use of projected incomplete trees for training. For a word that has no head words after projection, we enrich the projected structure by adding all possible words as its heads as long as the newly-added dependency does not cross any projected dependencies. In this way, the syntactic structure of a sentence becomes a parse forest (ambiguous labelings) instead of a single parse tree. During training, the objective is to maximize the mixed likelihood of manually labeled instances and projected instances with ambiguous labelings. Experimental results on benchmark data show that our method significantly outperforms a strong baseline supervised parser and previous syntactic projection methods.

## 2 Syntax Projection

Given an input sentence $\mathbf{x} = w_0 w_1 ... w_n$, a dependency tree is $\mathbf{d} = \{(h, m) : 0 \leq h \leq n, 0 < m \leq n\}$, where $(h, m)$ indicates a directed arc from the *head* word $w_h$ to the *modifier* $w_m$, and $w_0$ is an artificial node linking to the root of the sentence.

Syntax projection aims to project the dependency tree $\mathbf{d}^s$ of a source-language sentence $\mathbf{x}^s$ into the dependency structure of its target-language translation $\mathbf{x}$ via word alignments $\mathbf{a}$, where a word alignment $a_i = z$ means the target-side word $w_i$ is aligned into the source-side word $w_z^s$, as depicted in Figure 1(a) and Figure 1(b). For simplicity, we avoid one-to-many alignments by keeping the one with highest



(a) Source tree and word alignments (b) Projected incomplete tree (c) Forest (ambiguous labelings)

Figure 1: Illustration of syntax projection from English to Chinese with a sentence fragment. The two Chinese auxiliary words, "过$_3$" (past tense marker) and "的$_4$" (relative clause marker), are not aligned to any English words.

marginal probability when the target word is aligned to multiple source words. We first introduce a simple syntax projection approach based on DCA (Hwa et al., 2005), and then propose two extensions to handle parsing and aligning errors and cross-lingual syntactic divergences.

**Projection with DCA**. If two target words $w_i$ and $w_j$ are aligned to two different source words $w^s_{a_i}$ and $w^s_{a_j}$, and the two words compose a dependency in the source tree $(a_i, a_j) \in \mathbf{d}^s$, then add a dependency $(i, j)$ into the projected syntactic structure. For example, as shown in Figure 1(a), the two Chinese words "做$_2$" and "事$_5$" are aligned to the two English words "did$_3$" and "things$_1$", and the dependency "things$_1 \curvearrowright$did$_3$" is included in the source tree. Therefore, we project the dependency into the target side and add a dependency "做$_2 \curvearrowleft$事$_5$" into the projected structure, as shown in Figure 1(b). An obvious drawback of DCA is that it may produce many wrong dependencies due to the errors in the automatic source-language parse trees and word alignments. Even with manual parse trees and word alignments, syntactic divergences between languages can also lead to projection errors.

**Pruned with target-side marginals**. To overcome the weakness of DCA, we propose to use target-side marginal probabilities to constrain the projection process and prune obviously bad projections. We train a probabilistic parser on an existing target-side treebank. For each projected dependency, we compute its marginal probability with the target parser, and prune it off the projected structure if the probability is below a *pruning threshold* $\lambda_p$. Our study shows that dependencies with very low marginal probabilities are mostly wrong (Figure 2).

**Supplemented with target-side marginals**. To further improve the quality of projected structures, we add dependencies with high marginal probabilities according to the target parser. Specifically, if a target word $w_j$ obtain a head word $w_i$ after projection, and if another word $w_k$ has higher marginal probability than a *supplement threshold* $\lambda_s$ to be the head word of $w_j$, then we also add the dependency $(k, j)$ into the projected structure. In other words, we allow one word to have multiple heads so that the projected structure can cover more correct dependencies.

**From incomplete tree to forest**. Some words in the target sentence may not obtain any head words after projection due to incomplete word alignments or the pruning process, which leads to incomplete parse trees after projection. Also, some words may have multiple head words resulting from the supplement process. To handle these issues, we first convert the projected structures into parse forests, and then propose a generalized training technique based on ambiguous labelings to make use of the projected instances. Specifically, if a word does not have head words after projection, we simply add into the projected structure all possible words as its candidate heads as long as the newly-added dependency does not cross any projected dependencies, as illustrated in Figure 1(c). We introduce three new dependencies to compose candidate heads for the unattached word "过$_3$". Note that it is illegal to add the dependency "我$_1 \curvearrowright$过$_3$" since it would cross the projected dependency "做$_2 \curvearrowleft$事$_5$".

## 3 Dependency Parsing with Ambiguous Labelings

In parsing community, two mainstream methods tackle the dependency parsing problem from different perspectives but achieve comparable accuracy on a variety of languages. Graph-based methods view the problem as finding an optimal tree from a fully-connected directed graph (McDonald et al., 2005; McDonald and Pereira, 2006; Carreras, 2007; Koo and Collins, 2010), while transition-based methods try to find a highest-scoring transition sequence that leads to a legal dependency tree (Yamada and Matsumoto, 2003; Nivre, 2003; Zhang and Nivre, 2011).

### 3.1 Graph-based Dependency Parser (GParser)

We adopt the graph-based paradigm because it allows us to elegantly derive our CRF-based probabilistic parser, which is required to compute the marginal probabilities of dependencies and likelihood of both manually labeled data and unannotated bitext with ambiguous labelings. The graph-based method factors the score of a dependency tree into scores of small subtrees $\mathbf{p}$.

$$Score(\mathbf{x}, \mathbf{d}; \mathbf{w}) = \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, \mathbf{d}) = \sum_{\mathbf{p} \subseteq \mathbf{d}} Score(\mathbf{x}, \mathbf{p}; \mathbf{w}) \tag{1}$$

We adopt the second-order model of McDonald and Pereira (2006) as our core parsing algorithm,[1] which defines the score of a dependency tree as:

$$Score(\mathbf{x}, \mathbf{d}; \mathbf{w}) = \sum_{\{(h,m)\} \subseteq \mathbf{d}} \mathbf{w}_{dep} \cdot \mathbf{f}_{dep}(\mathbf{x}, h, m) + \sum_{\{(h,s),(h,m)\} \subseteq \mathbf{d}} \mathbf{w}_{sib} \cdot \mathbf{f}_{sib}(\mathbf{x}, h, s, m) \tag{2}$$

where $\mathbf{f}_{dep}(\mathbf{x}, h, m)$ and $\mathbf{f}_{sib}(\mathbf{x}, h, s, m)$ are feature vectors corresponding to two kinds of subtree; $\mathbf{w}_{dep/sib}$ are the feature weight vectors; the dot product gives the scores contributed by the corresponding subtrees. We adopt the state-of-the-art syntactic features proposed in Bohnet (2010).

### 3.2 Probabilistic CRF-based GParser

Previous work on dependency parsing mostly adopts linear models and online perceptron training, which lack probabilistic explanations of dependency trees and likelihood of the training data. Instead, we build a log-linear CRF-based probabilistic dependency parser, which defines the probability of a dependency tree as:

$$p(\mathbf{d}|\mathbf{x}; \mathbf{w}) = \frac{exp\{Score(\mathbf{x}, \mathbf{d}; \mathbf{w})\}}{Z(\mathbf{x}; \mathbf{w})}; \quad Z(\mathbf{x}; \mathbf{w}) = \sum_{\mathbf{d}' \in \mathcal{Y}(\mathbf{x})} exp\{Score(\mathbf{x}, \mathbf{d}'; \mathbf{w})\} \tag{3}$$

where $Z(\mathbf{x})$ is the normalization factor and $\mathcal{Y}(\mathbf{x})$ is the set of all legal dependency trees for $\mathbf{x}$.

### 3.3 Likelihood and Gradient of Training Data with Ambiguous Labelings

Traditional CRF models assume one gold-standard label for each training instance, which means each sentence is labeled with a single parse tree in the case of parsing. To make use of projected instances with ambiguous labelings, we propose to use a generalized training framework which allows a sentence to have multiple parse trees (forest) as its gold-standard reference (Täckström et al., 2013). The goal of the training procedure is to maximize the likelihood of the training data, and the model is updated to improve the probabilities of parse forests, instead of single parse trees. In other words, the model has the flexibility to distribute the probability mass among the parse trees inside the forest, as long as the probability of the forest improves. In this generalized framework, a traditional instance labeled with a single parse tree can be regarded as a special case that the forest contains only one parse tree.

The probability of a sentence $\mathbf{x}$ with ambiguous labelings $\mathcal{F}$ is defined as the sum of probabilities of all parse tree $\mathbf{d}$ contained in the forest $\mathcal{F}$:

$$p(\mathcal{F}|\mathbf{x}; \mathbf{w}) = \sum_{\mathbf{d} \in \mathcal{F}} p(\mathbf{d}|\mathbf{x}; \mathbf{w}) \tag{4}$$

[1] Higher-order models of Carreras (2007) and Koo and Collins (2010) can achieve a little bit higher accuracy, but suffer from higher time cost of $O(n^4)$ and system complexity. Our method is applicable to the third-order model.

| | Train | Dev | Test |
|---|---|---|---|
| PTB | 39,832 | 1,346 | 2416 |
| CTB5 | 16,091 | 803 | 1,910 |
| CTB5X | 18,104 | 352 | 348 |
| Bitext | 0.9M | – | – |

Table 1: Data sets (in sentence number).

Suppose the training data set is $\mathcal{D} = \{(\mathbf{x}_i, \mathcal{F}_i)\}_{i=1}^{N}$. Then the log likelihood of $\mathcal{D}$ is:

$$\mathcal{L}(\mathcal{D}; \mathbf{w}) = \sum_{i=1}^{N} \log p(\mathcal{F}_i | \mathbf{x}_i; \mathbf{w}) \tag{5}$$

Then we can derive the partial derivative of the log likelihood with respect to $\mathbf{w}$:

$$\frac{\partial \mathcal{L}(\mathcal{D}; \mathbf{w})}{\partial \mathbf{w}} = \sum_{i=1}^{N} \left( \sum_{\mathbf{d} \in \mathcal{F}_i} \tilde{p}(\mathbf{d}|\mathbf{x}_i, \mathcal{F}_i; \mathbf{w}) \mathbf{f}(\mathbf{x}_i, \mathbf{d}) - \sum_{\mathbf{d} \in \mathcal{Y}(\mathbf{x}_i)} p(\mathbf{d}|\mathbf{x}_i; \mathbf{w}) \mathbf{f}(\mathbf{x}_i, \mathbf{d}) \right) \tag{6}$$

where $\tilde{p}(\mathbf{d}|\mathbf{x}_i, \mathcal{F}_i; \mathbf{w})$ is the probability of $\mathbf{d}$ under the space constrained by the parse forest $\mathcal{F}_i$:

$$\tilde{p}(\mathbf{d}|\mathbf{x}_i, \mathcal{F}_i; \mathbf{w}) = \frac{\exp\{Score(\mathbf{x}_i, \mathbf{d}; \mathbf{w})\}}{Z(\mathbf{x}_i, \mathcal{F}_i; \mathbf{w})}; \quad Z(\mathbf{x}_i, \mathcal{F}_i; \mathbf{w}) = \sum_{\mathbf{d} \in \mathcal{F}_i} \exp\{Score(\mathbf{x}_i, \mathbf{d}; \mathbf{w})\} \tag{7}$$

The first term in Eq. (6) is the model expectations in the search space constrained by $\mathcal{F}_i$, and the second term is the model expectations in the complete search space $\mathcal{Y}(\mathbf{x}_i)$. Since $\mathcal{Y}(\mathbf{x}_i)$ contains exponentially many legal dependency trees, direct calculation of the second term is prohibitive. Instead, we can use the classic Inside-Outside algorithm to efficiently compute the second term within $O(n^3)$ time complexity, where $n$ is the length of the input sentence. Similarly, the first term can be solved by running the Inside-Outside algorithm in the constrained search space $\mathcal{F}_i$.

### 3.4 Stochastic Gradient Descent (SGD) Training

With the likelihood gradients, we apply L2-norm regularized SGD training to iteratively learn the feature weights $\mathbf{w}$ for our CRF-based baseline and bitext-enhanced parsers. We follow the implementation in CRFsuite.[2] At each step, the algorithm approximates a gradient with a small subset of training examples, and then updates the feature weights. Finkel et al. (2008) show that SGD achieves optimal test performance with far fewer iterations than other optimization routines such as L-BFGS. Moreover, it is very convenient to parallel SGD since computation among examples in the same batch is mutually independent.

Once the feature weights $\mathbf{w}$ are learnt, we can parse the test data and try to find the optimal parse tree with the Viterbi decoding algorithm in $O(n^3)$ parsing time (Eisner, 2000; McDonald and Pereira, 2006).

$$\mathbf{d}^* = \arg\max_{\mathbf{d} \in \mathcal{Y}(\mathbf{x})} p(\mathbf{d}|\mathbf{x}; \mathbf{w}) \tag{8}$$

## 4 Experiments and Analysis

To verify the effectiveness of our proposed method, we carry out experiments on English-to-Chinese syntax projection, and aim to enhance our baseline Chinese parser with additional training instances projected from automatic English parse trees on bitext. For **monolingual treebanks**, we use Penn English Treebank (PTB) and Penn Chinese Treebank 5.1 (CTB5). For English, we follow the standard practice to split the data into training (sec 02-21), development (sec 22), and test (sec 23). For CTB5, we adopt the data split of (Duan et al., 2007). We convert the original bracketed structures into dependency

[2] http://www.chokkan.org/software/crfsuite/

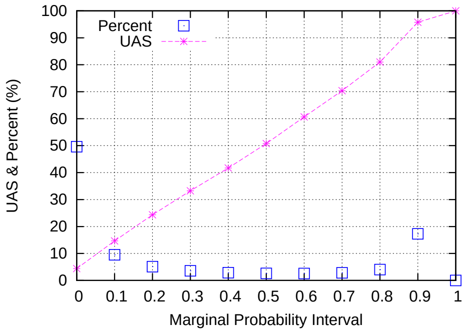


Figure 2: Distribution (Percent) and accuracy (UAS) of dependencies under different marginal probability interval for Chinese baseline parser on CTB5 development set. For example, $0.8$ at x-axis means the interval $[0.8, 0.9)$.

structures using Penn2Malt with its default head-finding rules. We build a CRF-based bigram part-of-speech (POS) tagger with the features described in (Li et al., 2012b), and produce POS tags for all train/development/test datasets and bitext (10-way jackknifing for training datasets). The tagging accuracy on test sets is $97.3\%$ on English and $94.0\%$ on Chinese.

To compare with the recent work on syntax projection of Jiang et al. (2010) who use a smaller test dataset, we follow their data split of CTB5 and use gold-standard POS tags during training and test. We refer to this setting as CTB5X.

For **bitext**, we collect a parallel corpus from FBIS news (LDC03E14, 0.25M sentence pairs), United Nations (LDC04E12, 0.62M), IWSLT2008 (0.04M), and PKU-863 (0.2M). After corpus cleaning, we obtain a large-scale bilingual parallel corpus containing 0.9M sentence pairs. We run the unsupervised BerkeleyAligner[3] (Liang et al., 2006) for 4 iterations to obtain word alignments. Besides hard alignments, we also make use of posterior probabilities to simplify one-to-many alignments to one-to-one as discussed in Section 2. Table 1 shows the data statistics.

For training both the baseline and bitext-enhanced parsers, we set the batch size to $100$ and run SGD until a maximum iteration number of $50$ is met or the change on likelihood of training data becomes too small. Since the number of projected sentences is much more than that of manually labeled instances (0.9M vs. 16K), it is likely that the projected data may overwhelm manually labeled data during training. Therefore, we adopt a simple corpus-weighting strategy. Before each iteration, we randomly sample 50K projected sentences and 15K manually labeled sentences from all training data, and run SGD to train feature weights using the sampled data. To speed up training, we adopt multi-thread implementation of gradient computations in the same batch. It takes about 1 day to train our bitext-enhanced parser for one iteration using a single CPU core, while using 24 CPU cores only needs about 2 hours.

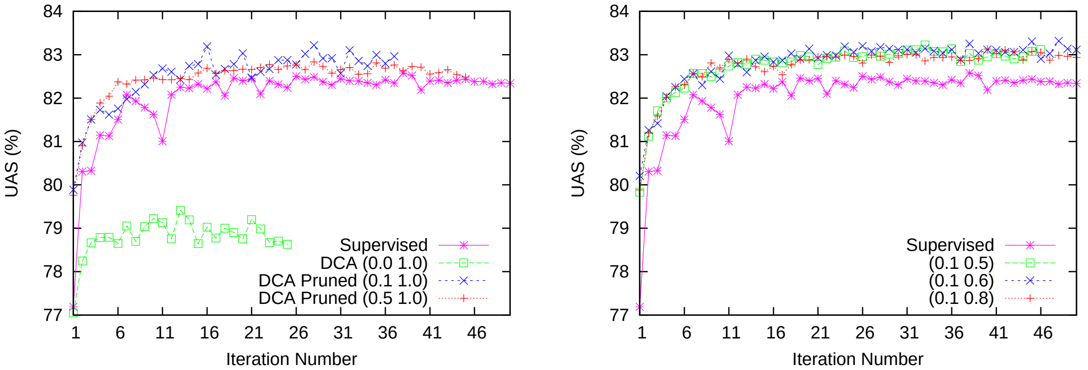
We measure parsing performance using unlabeled attachment score (UAS, percent of words with correct heads), excluding punctuation marks. For significance test, we adopt Dan Bikel's randomized parsing evaluation comparator (Noreen, 1989).[4]

### 4.1 Analysis on Marginal Probabilities

In order to gain insights for parameter settings of syntax projection, we analyse the distribution and accuracy of dependencies under different marginal probability interval. We train the baseline Chinese parser on CTB5 train set, and use the parser to produce the marginal probabilities of all dependencies for sentences in CTB5 development set. We discard all dependencies that have a marginal probability less than $0.0001$ for better illustration. Figure 2 shows the results, where we can see that UAS is roughly proportional to marginal probabilities. In other word, dependencies with higher marginal probabilities are more accurate. For example, dependencies with probabilities under interval $[0.8, 0.9)$ has a $80\%$ chance to be correct. From another aspect, we can see that $50\%$ of dependencies fall in probability

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

[4] http://www.cis.upenn.edu/~dbikel/software.html



(a) Parameters for DCA and DCA Pruned

(b) Parameters for DCA Pruned & Supplemented

Figure 3: Performance with different parameter settings of ($\lambda_p$ $\lambda_s$) on CTB5 development set.

interval $[0, 0.1)$, and such dependencies have very low accuracy ($4\%$). These observations are helpful for our parameter selection and methodology study during syntax projection.

### 4.2 Results of Syntax Projection on Development Dataset

We apply the syntax projection methods described in Section 2 to the bilingual text, and use the projected sentences with ambiguous labelings as additional training instances to train new Chinese parsers based on the framework described in Section 3. Figure 3 shows the UAS curves on development set with different parameters settings. The *pruning threshold* $\lambda_p$ (see Section 2) balances the quality and coverage of projection. Larger $\lambda_p$ leads to more accurate but fewer projections. The *supplement threshold* $\lambda_s$ (see Section 2) balances the size and oracle score of the projected forest. Smaller $\lambda_s$ can increase the oracle score of the forest by adding more dependencies with lower marginal probabilities, but takes the risk of making the resulted forest too ambiguous and weak to properly supervise the model during training. [5]

The *DCA* method corresponds to the results with $\lambda_p = 0.0$ and $\lambda_s = 1.0$. We can see that DCA largely decreases UAS compared with the baseline CRF-based parser. The reason is that although DCA projects many source-language dependencies to the target side ($44\%$ of target-language words obtain head words), it also introduces a lot of noise during projection.

*DCA pruned* with target-side marginals corresponds to the results with $\lambda_p > 0.0$ and $\lambda_s = 1.0$. Pruning with target-side marginals can clearly improve the projection quality by pruning out bad projections. When $\lambda_p = 0.1$, $31\%$ of target-language words obtain head words, and the model outperforms the baseline parser by $0.6\%$ at peak UAS. When $\lambda_p = 0.5$, the projection ratio decreases to $26\%$ and the improvement is $0.3\%$. Based on the results, we choose $\lambda_p = 0.1$ in later experiments.

Figure 3(b) presents the results of *DCA pruned & supplemented* with different $\lambda_s$. The supplement process adds a small amount of dependencies of high probabilities into the projected forest and therefore increases the oracle score, which provides the model with flexibility to distribute the probability mass to more preferable parse trees. We can see that although the peak UAS does not increase much, the training curve is more smooth and stable than that without supplement. Based on the results, we choose $\lambda_s = 0.6$ in later experiments.

### 4.3 Final Results and Comparisons on Test Dataset

Table 2 presents the final results on CTB5 test set. For each parser, we choose the parameters corresponding to the iteration number with highest UAS on development set. To further verify the usefulness of syntax projection, we also conduct experiments with self-training, which is known as a typical semi-supervised method. For the standard self-training, we use Chinese-side bitext with self-predicted parse trees produced by the baseline parser as additional training instances, which turns out to be hurtful to parsing performance. This is consistent with earlier results (Spreyer and Kuhn, 2009).

[5] Please note when $\lambda_p + \lambda_s >= 1$, $\lambda_s$ becomes useless. The reason is that if the probability of a projected dependency $(i, j)$ is larger $\lambda_p$, then no other word beside $w_i$ can have a probability larger than $\lambda_s$ of being the head word of $w_j$.

|  | UAS |
|---|---|
| Baseline Supervised Parser | 81.04 |
| Standard Self-training | 80.51 (-0.53) |
| Self-training with Ambiguous Labelings | 81.09 (+0.05) |
| DCA | 78.70 (-2.34) |
| DCA Pruned | 81.46 (+0.42 †) |
| DCA Pruned & Supplemented | 81.71 (+0.67 †) |

Table 2: UAS on CTB5 test set. † indicate statistical significance at confidence level of $p < 0.01$.

|  | Supervised | Bitext-enhanced |
|---|---|---|
| Jiang et al. (2010) | 87.15 | 87.65 (+0.50) |
| This work | 89.62 | 90.50 (+0.88 †) |

Table 3: UAS on CTB5X test set. † indicate statistical significance at confidence level of $p < 0.01$.

Then, we try a variant of self-training with ambiguous labelings following the practice in Täckström et al. (2013), and use a parse forest composed of dependencies of high probabilities as the syntactic structure of an instance. We can see that ambiguous labelings help traditional self-training, but still have no significant improvement over the baseline parser. Results in Table 2 indicate that our syntax projection method is able to project useful knowledge from source-language parse trees to the target-side forest, and then helps the target parser to learn effective features.

### 4.4 Comparisons with Previous Results on Syntax Projection on CTB5X

To make comparison with the recent work of Jiang et al. (2010), We rerun the process of syntax projection with CTB5X as the target treebank with the *DCA pruned & supplemented* method ($\lambda_p = 0.1$ and $\lambda_s = 0.6$).[6] Table 3 shows the results. Jiang et al. (2010) employ the second-order MSTParser of McDonald and Pereira (2006) with a basic feature set as their base parser. We can see that our baseline parser is much stronger than theirs. Even though, our approach leads to larger UAS improvement.

This work is different from theirs in a few aspects. First, the purpose of syntax projection in their work is to produce dependency/non-dependency instances which are used to train local classifiers to produce auxiliary features for MSTParser. In contrast, the outputs of syntax projection in our work are partial trees/forests where only reliable dependencies are kept and some words may receive more than one candidate heads. We directly use these partial structures as extra training data to learn model parameters. Second, their work measures the reliability of a projected dependencies only from the perspective of alignment probability, while we adopt a probabilistic parsing model and use target-side marginal probabilities to throw away bad projections, which turns out effective in handling syntactic non-isomorphism and errors in word alignments and source-side parses.

## 5 Related work

Cross-lingual annotation projection has been applied to many different NLP tasks to help processing resource-poor languages, such as POS tagging (Yarowsky and Ngai, 2001; Naseem et al., 2009; Das and Petrov, 2011) and named entity recognition (NER) (Fu et al., 2011). In another direction, much previous work explores bitext to improve monolingual NER performance based on bilingual constraints (Chen et al., 2010b; Burkett et al., 2010; Li et al., 2012a; Che et al., 2013; Wang et al., 2013).

Based on a universal POS tag set (Petrov et al., 2011), McDonald et al. (2011) propose to train delexicalized parsers on resource-rich language for parsing resource-poor language without use of bitext (Zeman and Resnik, 2008; Cohen et al., 2011; Søgaard, 2011). Täckström et al. (2012) derive cross-lingual clusters from bitext to help delexicalized parser transfer. Naseem et al. (2012) propose selectively sharing to better explore multi-source transfer information.

[6] In the previous draft of this paper, we directly use the projected data with in previous subsection for simplicity, and find that UAS can reach 91.39% (+1.77). The reason is that the CTB5X test is overlapped with CTB5 train. We correct this mistake in this version.

Our idea of training with ambiguous labelings is originally inspired by the work of Täckström et al. (2013) on multilingual parser transfer for unsupervised dependency parsing. They use a delexicalized parser trained on source-language treebank to obtain parse forests for target-language sentences, and re-train a lexicalized target parser using the sentences with ambiguous labelings. Similar ideas of learning with ambiguous labelings are previously explored for classification (Jin and Ghahramani, 2002) and sequence labeling problems (Dredze et al., 2009).

## 6 Conclusions

This paper proposes a simple yet effective framework of soft cross-lingual syntax projection. We make use of large-scale projected structures as additional training instances to boost performance of supervised parsing models trained on full-set manually labeled treebank. Compared with previous work, we make two innovative contributions: 1) using the marginal probabilities of a target-side supervised parser to control the projection quality with the existence of parsing and aligning errors and cross-lingual syntax divergences; 2) adopting a new learning technique based ambiguous labelings to make use of projected incomplete dependency trees for model training. Experimental results on two Chinese datasets demonstrate the effectiveness of the proposed framework, and show that the bitext-enhanced parser significantly outperforms all baselines, including supervised parsers, semi-supervised parsers based on self-training, and previous syntax projection methods.

Our anonymous reviewers present many great comments, especially on the experimental section. We will improve this work accordingly and release an extended version of this paper at the homepage of the first author. Such extensions include: 1) further exploring source-language parsing probabilities and alignment probabilities to help syntax projection; 2) studying the effect of the scale of source/target treebank and bilingual text.

## Acknowledgments

The authors would like to thank Wanxiang Che and Jiang Guo for sharing their bilingual data, and our anonymous reviewers for their critical and insightful comments, which will certainly help our future work. This work was supported by National Natural Science Foundation of China (Grant No. 61373095, 61203314, 61373097).

## References

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING*, pages 89–97.

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

David Burkett, Slav Petrov, John Blitzer, and Dan Klein. 2010. Learning better monolingual models with unannotated bilingual text. In *Proceedings of CoNLL 2010*, pages 46–54.

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of EMNLP/CoNLL*, pages 141–150.

Wanxiang Che, Mengqiu Wang, Christopher D. Manning, and Ting Liu. 2013. Named entity recognition with bilingual constraints. In *Proceedings of NAACL 2013*.

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

Yufeng Chen, Chengqing Zong, and Keh-Yih Su. 2010b. On jointly recognizing and aligning bilingual named entities. In *Proceedings of ACL 2010*.

Wenliang Chen, Jun'ichi Kazama, Min Zhang, Yoshimasa Tsuruoka, Yujie Zhang, Yiou Wang, Kentaro Torisawa, and Haizhou Li. 2011. SMT helps bitext dependency parsing. In *EMNLP*.

Shay B. Cohen, Dipanjan Das, , and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of EMNLP*.

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.

Mark Dredze, Partha Pratim Talukdar, and Koby Crammer. 2009. Sequence learning from data with multiple labels. In *ECML/PKDD Workshop on Learning from Multi-Label Data*.

Xiangyu Duan, Jun Zhao, and Bo Xu. 2007. Probabilistic parsing action models for multi-lingual dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 940–946.

Jason Eisner. 2000. Bilexical grammars and their cubic-time parsing algorithms. In *Advances in Probabilistic and Other Parsing Technologies*, pages 29–62.

Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based, conditional random field parsing. In *Proceedings of ACL*, pages 959–967.

Ruiji Fu, Bing Qin, and Ting Liu. 2011. Generating chinese named entity data from a parallel corpus. In *Proceedings of IJCNLP 2011*, pages 264–272.

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

Kuzman Ganchev, Jo ao Graca, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Artifical Intellignece Research*.

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

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Boostrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3):311–325.

Wenbin Jiang, , and Qun Liu. 2010. Dependency parsing and projection based on word-pair classification. In *ACL*, pages 897–904.

Rong Jin and Zoubin Ghahramani. 2002. Learning with multiple labels. In *Proceedings of NIPS*.

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

Qi Li, Haibo Li, Heng Ji, Wen Wang, Jing Zeng, and Fei Huang. 2012a. Joint bilingual name tagging for parallel corpora. In *Proceedings of CIKM 2012*.

Zhenghua Li, Min Zhang, Wanxiang Che, and Ting Liu. 2012b. A separately passive-aggressive training algorithm for joint POS tagging and dependency parsing. In *COLING 2012*, pages 1681–1698.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *HLT-NAACL*.

Kai Liu, Yajuan Lü, Wenbin Jiang, and Qun Liu. 2013. Bilingually-guided monolingual dependency grammar induction. In *Proceedings of ACL*.

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

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*, pages 91–98.

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

Tahira Naseem, Benjamin Snyder, Jacob Eisentein, and Regina Barzilay. 2009. Multilingual part-of-speech tagging: two unsupervised approaches. *Journal of Artifical Intellignece Research*, 36(1):341–385.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of ACL*.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of IWPT*, pages 149–160.

Eric W. Noreen. 1989. *Computer-intensive methods for testing hypotheses: An introduction*. John Wiley & Sons, Inc., New York.

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

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

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of ACL 2011*, pages 682–686.

Kathrin Spreyer and Jonas Kuhn. 2009. Data-driven dependency parsing of new languages using incomplete and noisy training data. In *CoNLL*, pages 12–20.

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

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of NAACL*, pages 1061–1071.

Mengqiu Wang, Wanxiang Che, and Christopher D. Manning. 2013. Joint word alignment and bilingual named entity recognition using dual decomposition. In *Proceedings of ACL 2013*.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, pages 195–206.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proceedings of NAACL 2001*.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of IJCNLP 2008*.

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