Joint Learning of Constituency and Dependency Grammars by Decomposed Cross-Lingual Induction

Wenbin Jiang [1] and **Qun Liu** [1, 2] and **Thepchai Supnithi** [3]
[1]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology
Chinese Academy of Sciences, China
[2]ADAPT Centre, School of Computing, Dublin City University, Ireland
[3]National Electronics and Computer Technology Center, Thailand
jiangwenbin@ict.ac.cn

Abstract

Cross-lingual induction aims to acquire for one language some linguistic structures resorting to annotations from another language. It works well for simple structured predication problems such as part-of-speech tagging and dependency parsing, but lacks of significant progress for more complicated problems such as constituency parsing and deep semantic parsing, mainly due to the structural non-isomorphism between languages. We propose a *decomposed projection* strategy for cross-lingual induction, where cross-lingual projection is performed in unit of fundamental decisions of the structured predication. Compared with the *structured projection* that projects the complete structures, decomposed projection achieves better adaptation of non-isomorphism between languages and efficiently acquires the structured information across languages, thus leading to better performance. For joint cross-lingual induction of constituency and dependency grammars, decomposed cross-lingual induction achieves very significant improvement in both constituency and dependency grammar induction.

1 Introduction

For parsing of resource-rich languages, supervised learning on manual treebanks achieves the state-of-the-art [Collins, 1999; McDonald *et al.*, 2005; Petrov *et al.*, 2006; Koo and Collins, 2010]. Unsupervised methods utilizing raw text achieve progress [Klein and Manning, 2004; Bod, 2006; Headden III *et al.*, 2009; Spitkovsky *et al.*, 2013], but the performance is still far from applicable. Compared with unsupervised methods, cross-lingual induction usually achieves better performance, especially in structurally simple problems such as part-of-speech tagging and dependency parsing [Hwa *et al.*, 2005; Ganchev *et al.*, 2009; Smith and Eisner, 2009; McDonald *et al.*, 2011]. However, there is no significant progress in cross-lingual induction for more complicated tasks such as constituency parsing and semantic parsing.

For complicated syntactic or semantic paradigms, structured projection based on direct correspondence assumption [Hwa *et al.*, 2005] is hard to achieve promising performance due to the structural complicity and non-isomorphism between languages. However, the success in bilingual parsing [Burkett and Klein, 2008; Huang *et al.*, 2009] gives us the inspiration that, *appropriately leveraging* rather than *strictly obeying* the inner isomorphism between two languages may be critical to the successful cross-lingual induction of complicated structures. Structural projection, which directly project linguistic structures from one language to another, gives a strong assumption of isomorphism between the source and the target languages. The structural non-isomorphism between languages makes it hard to apply structural projection to the cross-lingual induction of complicated linguistic structures.

Most syntactic or semantic parsing models factorize the parsing procedure into a set of fundamental decisions. For example, a transition-based parser performs a series of transition operations to find a tree structure [Nivre and Scholz, 2004; Sagae and Lavie, 2005]. Using the fundamental decision as the unit of cross-lingual induction probably leads to better performance for cross-lingual induction. In this paper we design a more effective, decomposed projection strategy for joint cross-lingual induction of constituency and dependency grammars. We remodel the constituency parsing as a variant of transition-based parsing, where a transition operation determines the behavior of two neighbor phrase structures, including whether two neighbors could be merged, what non-terminal should they be merged as, and which kid will be the head. Since each decision contains the determination of the head kid, such model performs dependency parsing simultaneously. After the remodeling of the parsing procedure, a novel cross-lingual extraction algorithm is designed in order to acquire the training instances of transition decisions. This manner reduces the isomorphism assumption in cross-lingual induction, while guaranteeing the efficient acquirement of structural information necessary for structural prediction.

We experiment on joint cross-lingual induction of constituency and dependency grammars from English to Chinese. We first verify the effectiveness of the transition-based variant model for constituency parsing. On WSJ treebank, this model achieves accuracy comparable to the classic transition-based model. We use FBIS Chinese-English dataset as the bilingual corpus for cross-lingual induction. The joint constituency and dependency grammar induced by the decomposed strat-

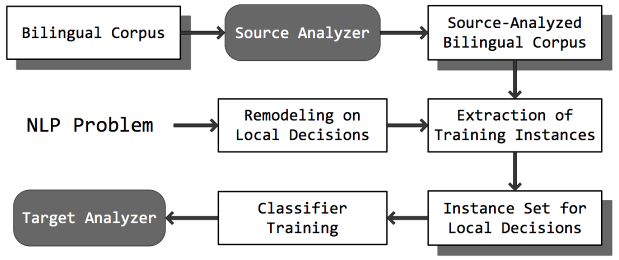
Bilingual Corpus
Source Analyzer
Source-Analyzed Bilingual Corpus
NLP Problem
Remodeling on Local Decisions
Extraction of Training Instances
Target Analyzer
Classifier Training
Instance Set for Local Decisions

Figure 1: The flowchart of the overall procedure for the decomposed cross-lingual induction.

egy achieves promising accuracy on constituency and dependency parsing, significantly outperforming previous work on cross-lingual or unsupervised grammar induction. Although it is not fair to compare with unsupervised models that utilize only raw text, cross-lingual induction is a more effective strategy to build an initial model if the resource-scarce language has bilingual corpus parallel with a resource-rich language. In future work, we will extend this approach to other complicated NLP problems such as deep semantic analysis.

2 Remodeling Constituency&Dependency Parsing

We factorize the joint constituency and dependency parsing into a set of decisions of the most probable transition operation between two neighbor constituencies. A transition operation, denoted as a tuple $\mathcal{T} = (\lambda, \alpha, \beta)$, indicating three aspects of decisions, whether two neighbors could be merged ($\lambda \in \{\mathbf{reduce}, \mathbf{separate}\}$), what non-terminal should they be merged as ($\alpha \in \mathbf{NT}$, the set of non-terminals), and which kid will be the head ($\beta \in \{\mathbf{left}, \mathbf{right}\}$). The first two aspects correspond to the decisions for constituency parsing, while the last one for dependency parsing. Note that α and β are undefined (denoted as $\emptyset$) when λ equals $\mathbf{separate}$.

Therefore, a transition operation (λ, α, β) can be divided into a constituency transition operation (λ, α) operation and a dependency transition operation (λ, β). There are $|\mathbf{NT}| + 1$ constituency transition operations, $(\mathbf{reduce}, \alpha)$ where α is enumerated in $\mathbf{NT}$, and $(\mathbf{separate}, \emptyset)$. For dependency transition there are three possible operations, $(\mathbf{reduce}, \mathbf{left})$, $(\mathbf{reduce}, \mathbf{right})$, and $(\mathbf{separate}, \emptyset)$. The probability of a transition operation is the multiplication of the probabilities of the corresponding two component transition operations:

$$\begin{aligned} p(\mathcal{T}|\mathcal{S}, \mathcal{C}_c, \mathcal{C}_d) &= p(\lambda, \alpha, \beta|\mathcal{S}, \mathcal{C}_c, \mathcal{C}_d) \\ &= p(\lambda, \alpha|\mathcal{S}, \mathcal{C}_c) \times p(\lambda, \beta|\mathcal{S}, \mathcal{C}_d) \end{aligned} \tag{1}$$

Here, $\mathcal{S}$ denotes the states in the transition-based parsing procedure, $\mathcal{C}_c$ and $\mathcal{C}_d$ denote the constituency and dependency classifiers. This parsing model could be seen as a variant of the transition-based method where the transition operation is determined by two component classifiers. The parser searches for the most probable parsing tree according to the formula (where the conditions for probabilities are omitted for simplicity):

Algorithm 1 K-beam transition-based parsing.

```
Input: sentence: $\mathbf{x}$, beam size: $k$, classifiers: $\mathcal{C}_c$ and $\mathcal{C}_d$
insert InitState($\mathbf{x}$) into queue
for $i \leftarrow 1 .. 2|\mathbf{x}| - 1$ do                ▷ $2|\mathbf{x}| - 1$ iterations
    buf $\leftarrow \emptyset$
    for $\mathcal{S} \in$ queue do                        ▷ for each state $\mathcal{S}$
        for $\mathcal{T} \in$ GetActions($\mathcal{S}$) do
            insert NextState($\mathcal{S}$, $\mathcal{T}$) into buf
    queue $\leftarrow$ $k$ best states in buf
Output: the tree derived from the best state in queue
```

$$\begin{aligned} \mathbf{y}(\mathbf{x}) &= \arg\max_{\mathbf{y}} \Bigg(\sum_{\mathcal{D},\mathbf{s.t.}\mathcal{D}(\mathbf{x})=\mathbf{y}} \prod_{\mathcal{T}\in\mathcal{D}} p(\mathcal{T}) \Bigg) \\ &\approx \arg\max_{\mathbf{y}} \Bigg(\max_{\mathcal{D},\mathbf{s.t.}\mathcal{D}(\mathbf{x})=\mathbf{y}} \prod_{\mathcal{T}\in\mathcal{D}} p(\mathcal{T}) \Bigg) \end{aligned} \tag{2}$$

Here $\mathcal{D}$ is a derivation, that is, a sequence of transition operations lead to the candidate tree $\mathbf{y}$. A summation operation is needed to accumulate across all the derivations corresponding to the same tree, but it can be estimated by a max operator to facilitate approximate decoding.

Following previous work on transition-based parsing, a binarization method is adopted in the treebank processing and the parsing procedure. The binarization is done in a strict order from the head to the right and then to the left. Furthermore, we shrink each single-branch path into one node, in order to generate strict binary trees. The determination of transition operations, therefore, needs only consider two neighbor constituencies. It is easy to extract the normal trees from the binary trees with the asterisks as indications.

Training instances for constituency and dependency transitions are extracted automatically from the binarized treebank, and then used to train the classifiers. Both classifiers share the same set of feature templates, considering the constituency and lexical information of the two neighbor phrase structures and their context, as shown in Table 1. The feature templates considering constituency information are adopted from classic transition-based methods [Sagae and Lavie, 2005], and those considering dependency information are adopted from first-order maximum spanning tree methods [McDonald *et al.*, 2005].

The variant transition-based model is also different from the traditional models in decoding. In the initial state before decoding procedure, each word and its part-of-speech tag form a minimal subtree. The most intuitive decoding strategy is to scan the subtree sequence from left to right, and perform transition operations to each pair of adjacent subtrees according to predication of the classifiers. If predicated as $(\mathbf{separate}, \emptyset, \emptyset)$, no reduction operation will be conducted on the two subtrees and the current considering window moves forward. Otherwise, the two subtrees will be reduced into a larger subtree according to the predicated operation. This procedure continues iteration by iteration until a complete tree structure obtained. As described by previous work, it is hard for such a deterministic parsing algorithm to

Type	Feature Templates			
NT-related	lhww ○ lhwt ○ rhww ○ rhwt	lhww ○ lhwt ○ rhww	lhww ○ lhwt ○ rhwt	lhww ○ rhww ○ rhwt
	lhwt ○ rhww ○ rhwt	lhww ○ rhww	lhwt ○ rhwt	lsym ○ lhww ○ rsym ○ rhww
	lsym ○ lhww ○ rsym	lsym ○ lhww ○ rhww	lsym ○ rsym ○ rhww	lhww ○ rsym ○ rhww
	lsym ○ lhwt ○ rsym ○ rhwt	lsym ○ lhwt ○ rsym	lsym ○ lhwt ○ rhwt	lsym ○ rsym ○ rhwt
	lhwt ○ rsym ○ rhwt	lsym ○ rsym	elw ○ elt ○ erw ○ ert	elw ○ elt ○ erw
	elw ○ elt ○ ert	elw ○ erw ○ ert	elt ○ erw ○ ert	lsym ○ elw ○ rsym ○ erw
	lsym ○ elw ○ rsym	lsym ○ rsym ○ erw	lsym ○ elt ○ rsym ○ ert	lsym ○ elt ○ rsym
	lsym ○ rsym ○ ert			
Lexical	pw	pt	pw ○ pt	cw
	ct	cw ○ ct	pw ○ pt ○ cw ○ ct	pw ○ pt ○ cw
	pw ○ pt ○ ct	pw ○ cw ○ ct	pt ○ cw ○ ct	pw ○ cw
	pt ○ ct	pw ○ ct	pt ○ cw	pt ○ pt-1 ○ ct ○ ct-1
	pt ○ pt+1 ○ ct ○ ct+1	pt ○ pt+1 ○ ct ○ ct-1	pt ○ pt-1 ○ ct ○ ct+1	pt ○ pt-1 ○ ct-1
	pt ○ pt-1 ○ ct+1	pt ○ pt+1 ○ ct-1	pt ○ pt+1 ○ ct+1	pt-1 ○ ct ○ ct-1
	pt-1 ○ ct ○ ct+1	pt+1 ○ ct ○ ct-1	pt+1 ○ ct ○ ct+1	pt ○ ct ○ ct-1
	pt ○ ct ○ ct+1	pt ○ pt-1 ○ ct	pt ○ pt+1 ○ ct	

Table 1: Feature templates for transition-based parsing model. lhww/lhwt: the word/POS of the head word of the left constituency; rhww/rhwt: the word/POS of the head word of the right constituency; lsym/rsym: the non-terminal symbol of the left/right constituency; elw/wlt: the word/POS of the token on the left of the left constituency; erw/ert: the word/POS of the token on the right of the right constituency; pw/pt: the word/POS of the supposed head; cw/ct: the word/POS of the supposed modifier; pt-1/pt+1: the POS to the left/right of the supposed head; ct-1/ct+1: the POS to the left/right of the supposed modifier. Besides the original features generated according to the templates, the enhanced features with distance (between the heads the two constituencies) as postfixes are also used in training and decoding.

achieve very high accuracy due to serious error-propagation.

An optimized strategy is the best-first transition-based parsing. From the current state, the pair of adjacent subtrees with the highest reducing probability is chosen and reduced according to the predicted operation so as to arrive at the next state. Such a best-first procedure iterates until there is only one tree remained in the state. Based on the best-first algorithm we further introduce the k-beam searching strategy. At any time t, there are at most k best states maintained in a queue. Each of these states pops out of the queue and generates its own succeeding states individually, then the k best ones of all the succeeding states are reserved for the startup of the next iteration at time $t + 1$. Algorithm 1 gives the pseudo-code for k-beam transition-based parsing.

The training of the classifiers for transition predication is straightforward. We first extract two sets of transition instances from the annotated treebank, then train classifiers on the instance sets. Since predication probabilities are needed by k-beam transition-based parsing, the classifier should give probabilistic predications rather than a single predicated choice.

3 Cross-Lingual Extraction of Transition Instances

Given a bilingual corpus with lexical alignment between each pair of sentences, the source sentence and the target sentence, the transition operations indicated by the syntactic tree of the source sentence (source tree for short) can be projected onto the target sentence across the alignment. Such a cross-lingual projection procedure extracts a set of transition instances, which is then used to train the target language parser just as in the monolingual scenario. Similar to the treebank binarization in the supervised learning situation, a source tree also needs binarization before cross-lingual induction. Considering the syntactic non-isomorphism between two languages, we further conduct exhaustive binarization for constituencies with more than two kids so as to generate a binary forest (source forest for short).

We first describe the cross-lingual extraction of transition instances. For two adjacent spans in the target sentence (target span for short), x_l and x_r, if they exactly correspond to two constituencies in the source forest (source constituency for short), y_l and y_r, a transition instance can be extracted for x_l and x_r according to the relationship of y_l and y_r in the source forest. Specifically, if y_l and y_r form a larger constituency α, the transition operation of the cross-lingually induced instances is $(\mathbf{reduce}, \alpha, \beta)$, where the unspecified β indicates that the head child is y_l or y_r; If they can not form a constituency, the transition operation of the induced instances will be $(\mathbf{separate}, \emptyset, \emptyset)$. According to the transition operation, two transition instances can be extracted for constituency transition and dependency transition, respectively.

It relies on the lexical alignment to find the correspondence between source constituencies and target spans. The determination of whether a target span corresponds to a source constituency across the alignment is similar to the rule extraction in statistical machine translation. The span x exactly corresponds to the constituency y, if and only if each word in x is aligned to words inside y, and each word covered by y is aligned to words inside x. The head word of the span x is the word aligned with the head word of the constituency y.

To alleviate the errors in the unsupervised learned word alignment, we adopt the probabilistic word alignment $\mathcal{A}$ to improve the cross-lingual extraction of transition instances. In probabilistic word alignment, a word in the source sentence is aligned to all the words in the target sentence, of course, with different probabilities. For example, $p(i, j)$ indicates the probability that the source word i is aligned to the target word j. With probabilistic word alignment, the correspondence between the target span x and the source constituency y is not a binary value, but a probability indicating the cross-lingual equivalent degree between x and y, which can be calculated by:

$$\begin{aligned} p(x, y|\mathcal{A}) &= p(x|y, \mathcal{A}) \times p(y|x, \mathcal{A}) \\ &= \frac{\sum_{i \in x, j \in y} \mathcal{A}(i, j)}{\sum_{i \in x} \mathcal{A}(i, j)} \times \frac{\sum_{i \in x, j \in y} \mathcal{A}(i, j)}{\sum_{j \in y} \mathcal{A}(i, j)} \end{aligned} \quad (3)$$

For a pair of adjacent target spans, x_l and x_r, and a pair of (not necessary adjacent) source constituencies, y_l and y_r, a candidate instance can be extracted with a probability:

$$p(x_l, x_r, y_l, y_r|\mathcal{A}) = p(x_l, y_l|\mathcal{A}) \times p(x_r, y_r|\mathcal{A}) \quad (4)$$

Training instances are selected from the enumerated candidates according to the probabilities. Specifically, for the situations where the pair of source constituencies form a larger source constituency, only the candidate instance with the highest probability can be reserved for each pair of constituencies. From these reserved instances, the top $N - 1$ best candidates are selected as the final training instances, where N denotes the length of the target sentence. The head word of the span x is the word aligned with the head word of the constituency y with the highest probability.

4 Related Work

Work on learning or improving NLP models in a multilingual manner includes at least three groups, bilingual learning, universal grammar, and cross-lingual induction. In decades, bilingual learning achieves progress in constituency and dependency parsing [Burkett and Klein, 2008; Huang *et al.*, 2009]. This strategy supposes that there are some manual annotations for each language, then improves the parameter leaning for one language or both languages resorting to some kinds of bilingual constraints. Universal grammar is an interesting methodology, there are some promising researches in recent years [McDonald *et al.*, 2013]. It supposes that there are some degree of universal isomorphism among the languages in the world, and designs parallel grammar or parallel annotated corpora for multiple languages. Cross-lingual induction aims to learn an NLP model for one language resorting to the annotations from another language. It achieves significant progress on part-of-speech tagging and dependency parsing [Hwa *et al.*, 2005; Ganchev *et al.*, 2009; Smith and Eisner, 2009; McDonald *et al.*, 2011], but still performs poorly on structurally complicated problems such as constituency parsing.

There are already some investigations on improving cross-lingual induction by alleviating the isomorphism assumption. For dependency parsing, quasi-synchronous grammar can improve the performance of cross-lingual projection [Smith and Eisner, 2009]. For dependency parsing and part-of-speech tagging, using the dependency edge or part-of-speech tag as the unit of cross-lingual projection is a natural and reasonable strategy for these specific problems [Das and Petrov, 2011]. For constituency parsing, researchers achieved higher accuracy for cross-lingual induction when allowing a subsequence in the target language sentence to correspond to an incomplete treelet in the source tree [Jiang *et al.*, 2011]. However, these methods are either suitable for only some specific problems, or limited in their ability to alleviate the isomorphism assumption.

Treebank	Training	Developing	Testing
WSJ	02-21	22	23
CTB	1-270 400-931 1001-1151	301-325	271-300

Table 2: Data partitioning for WSJ and CTB, in unit of section.

Compared to previous work, this work proposes for cross-lingual induction a more universal, decomposed projection strategy, where cross-lingual induction is performed in unit of fundamental decisions of the structural predication. Admitting the cross-lingual consistency in the level of fundamental decisions while abandoning the isomorphism in the level of linguistic structures, it can alleviate the isomorphism assumption to the maximum degree. For more structurally complicated problems, joint induction of constituency and dependency grammar, decomposed induction achieves significant better performance over previous work by appropriately remodeling of the NLP problem and effective cross-lingual extraction of training instances. This strategy is easier to be extended to complicated NLP problems such as deep semantic parsing.

5 Experiments

We first evaluate the performance of the remodeled transition-based parsing algorithm on the Wall Street Journal Treebank (WSJ) [Marcus *et al.*, 1993], where we use the balanced F-measure as the accuracy for constituency parsing, and the head attachment precision for dependency parsing. Then we verify the effectiveness of decomposed cross-lingual grammar induction, with experiments from English to Chinese using the FBIS Chinese-English dataset as the bilingual corpus. The accuracy of the induced grammar is evaluated on some portions of the Penn Chinese Treebank (CTB) [Xue *et al.*, 2005]. Especially for constituency grammar, following the previous work of unsupervised constituency parsing, we evaluate the induced grammar on the subsets of CTB 1, CTB 2 and CTB 5, which contain no more than 10 or 40 words after the removal of punctuation. The gold-standard POS tags are directly used for testing. The evaluation for unsupervised parsing differs slightly from the standard metrics, it ignores the multiplicity of brackets, brackets of span one, and the bracket labels.

Two training instance sets are extracted from the banalized trees in WSJ for constituency and dependency transition decisions. For both training sets, the proportions of instances

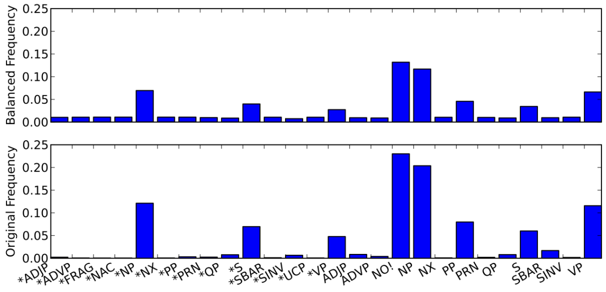

Figure 2: Frequencies of the transition instances with different α (non-terminals), including the original distribution (below) and the balanced distribution (upper). Items with original frequency less than 0.05% are omitted for simplicity.

System	Cons. F%/Dep. P%
Constituency Parsing	
[Tsuruoka and Tsujii, 2005]	85.9
[Sagae and Lavie, 2005]	86.0
[Zhu *et al.*, 2013]	91.3
Our Model	85.3
Dependency Parsing	
[Yamada and Matsumoto, 2003]	90.3
[Nivre and Scholz, 2004]	87.3
[Zhang and Clark, 2008]	92.1
[Huang and Sagae, 2010]	92.1
Our Model	89.4

Table 3: The performance of different transition-based parsing algorithms on WSJ, compared with previous work on transition-based parsing. Note that the latest transition-based parsers achieve significant improvement by using dynamic programming, model combination, or other resources.

with different labels are very unbalanced. It is even worse for the training instance set for constituency transition decisions. Figure 2 shows the statistics of transition instances according to different α (non-terminal) on WSJ. We can find that the freq of different groups of instances exhibits a very unbalanced distribution.

It is relatively easy to make a balance between different labels in training instances for dependency transition decisions. A great deal of instances with label $(\mathbf{separate}, \emptyset)$ are obtained due to the exhaustive enumeration, but we can randomly select k_d instances out of them, where k_d is simply set as the larger one between the counts of the other two kinds of instances. For the training instance set for constituency transition decisions, it is much harder to balance between labels, since there are much more labels as well as much larger diversities between the proportions of different labels. An intuitive strategy is to duplicate the instances with minority labels to balance with the instances with majority labels. If the instances with a specific label are less than the averaged count (across all the labels), we simply duplicate them to the averaged count. Figure 2 also shows the balanced distribution of instances with different labels. We find that the transition-based model together with the simple balancing strategies already leads to promising performance.

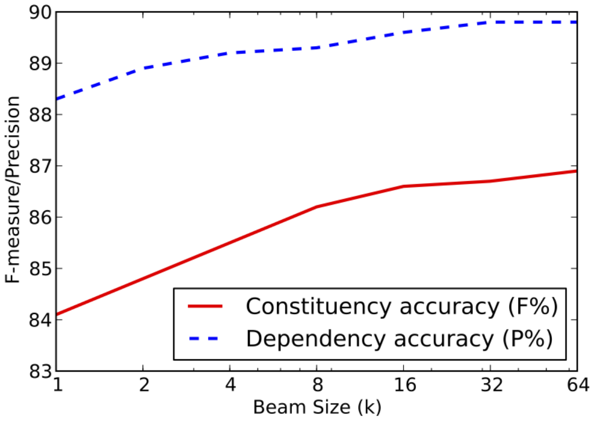

Figure 3: Developing curve of the k-beam transition-based parsing algorithm.

Alignment	Cons. F%	Dep. P%
Deterministic alignment	52.7	58.0
Probabilistic alignment	53.4	58.8

Table 4: The performance of the cross-lingually induced joint constituency and dependency grammar on the test set of CTB 5.

After the extraction of the training instances, two classifiers are trained with the maximum entropy toolkit by Zhang [1]. We set the gaussian prior as 1.0, the cutoff threshold as 0 (without cutoff), and the maximum training iteration as 100, while leaving other parameters as default values. With the transition decisions given by the classifiers, joint constituency and dependency parsing can be conducted by the transition-based algorithms described before. For the k-beam transition-based parsing algorithm, the developing curve is shown in Figure 3, where less improvement can be obtained with k larger than 16. Table 3 shows the performance of this algorithm on WSJ, compared with previous work on transition-based parsing. We find that the accuracy of the k-beam transition-based parsing is comparable to the state-of-the-art transition-based parsers, although without using complicated syntactic features.

For the cross-lingual induction of transition instances, we perform word alignment by running GIZA++ [Och, 2003] to automatically obtain the lexical correspondence information. The probabilistic alignment for each sentence pair is generated by summation and normalization of the k-best GIZA++ results, where k is set as 10 in our experiments. Training instance sets for constituency and dependency transition are extracted according to the principle as described before. Table 4 shows the accuracy of the grammars cross-lingually induced based on deterministic and probabilistic alignment. We find

[1] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Previous Work		F/P(%)	Our Work
Constituency Grammar Induction			
[Klein and Manning, 2004]	♢	46.7	57.3 (**+10.6**)
[Bod, 2006]	♢	47.2	57.3 (**+10.1**)
[Seginer, 2007]	△	38.0	52.5 (**+14.5**)
[Jiang *et al.*, 2011]	△	49.2	52.5 (**+3.3**)
[Parikh *et al.*, 2014]	▽	42.2	52.5 (**+10.3**)
Dependency Grammar Induction			
[Klein and Manning, 2004]	♢	55.2	59.7 (**+4.5**)
[Hwa *et al.*, 2005]	♡	53.9	57.5 (**+3.6**)
[McDonald *et al.*, 2011]	♣	49.3	58.8 (**+9.5**)
[Naseem *et al.*, 2012]	♣	51.2	58.8 (**+7.6**)
[Spitkovsky *et al.*, 2013]	♣	58.4	58.8 (**+0.4**)
	♠	52.5	58.8 (**+6.3**)

Table 5: The performance of the cross-lingually induced grammar on CTB compared with previous work on constituency and dependency grammar induction. ♢: sentences $\leq$ 10 words from CTB 1 after the removal of punctuation; △/▽: sentences $\leq$ 40 words from CTB 5 or the last 20% of CTB 5 after the removal of punctuation; ♡: the test set of CTB 2 defined by Hwa et al. [2005]; ♣/♠: the Chinese section of CoNLL test set of 2006/2007. Although we have not obtained the datasets of CoNLL06 and CoNLL07, we give our results on the test set of CTB 5 to make a rough comparison.

that the probabilistic alignment brings obvious improvement over the baseline using deterministic alignment. On both constituency and dependency accuracy, the final grammar significantly outperforms previous work on unsupervised or cross-lingual grammar induction especially on long sentences, as shown in Table 5. It is not fair for cross-lingual induction to compare with unsupervised models that utilize only raw text. However, it is a more effective strategy to build an initial model, or a good initialization for unsupervised models, if the resource-scarce language that we focus on has bilingual corpus parallel with a resource-rich language.

Considering the difference between the annotation styles of the source language treebank and the target language testing set, it can be estimated that the performance of the cross-lingual grammar induction is also limited by the consistency between the annotation styles. Since there are no complete trees generated in the cross-lingual grammar induction procedure, we conduct an initial experiment at the level of classification instances, as shown in Figure 4. The constituency transition instance sets are used since the non-isomorphism between constituency grammars is much larger than between dependency grammars. Indicated by the training curves, a classifier performs well on the developing set corresponding to the training set (even for the cross-lingually induced instances), but performs significantly worse on a different developing set. This initial experiment inspires us to investigate the difference between annotation styles of the test set and the induced grammar. It would be a promising direction for future improvement to address and tackle the difference and relationship between annotation styles.

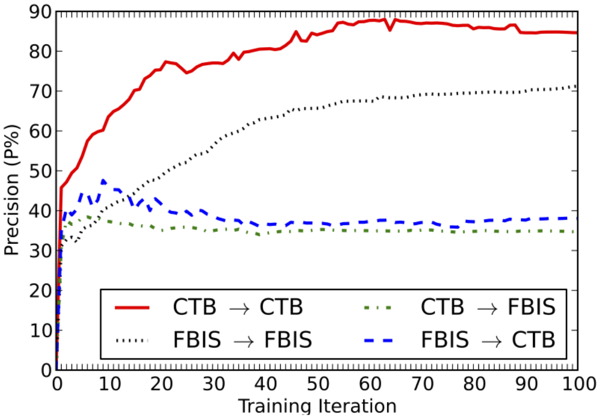

Figure 4: Performance curve of the constituency classifiers on developing sets. X→Y indicates that the classifier trained on instances extracted from the training portion of X is evaluated on the instances extracted from the developing portion of Y.

6 Conclusion and Future Work

This paper describes a decomposed projection strategy for cross-lingual grammar induction. On joint cross-lingual induction of constituency and dependency grammars from English to Chinese, the decomposed strategy significantly outperforms previous work on cross-lingual or unsupervised grammar induction. This validates that, for an NLP problem, if there is an appropriate remodeling where the fundamental decisions not only support the efficient derivation of a good analysis, but also facilitate the cross-lingual extraction of decision instances, better models can be obtained by decomposed cross-lingual induction.

This work can be improved in the future in two aspects. First, although the probabilistic lexical alignment improves the cross-lingual induction to some degree, it is still restricted by the performance of lexical alignment. Joint lexical alignment and grammar induction may be more effective. Second, a pipeline manner is adopted in the current method, that is, cross-lingual projection of instances followed by offline training of the classifiers, which is inconvenient for utilization of complicated syntactic features. It is hopeful to try a synchronous manner that performs cross-lingual projection and classifier training incrementally, in order to leverage more complicated features.

Acknowledgments

Jiang and Liu are supported by National Natural Science Foundation of China (Contract 61202216). Liu is partially supported by the Science Foundation Ireland (Grant 12/CE/I2267 and 13/RC/2106) as part of the ADAPT Centre at Dublin City University. The authors are also partially supported by Sino-Thai Scientific and Technical Cooperation (No. 60-625J). We sincerely thank the anonymous reviewers for their insightful comments.

References

[Bod, 2006] Rens Bod. An all-subtrees approach to unsupervised parsing. In *Proceedings of the COLING-ACL*, 2006.

[Burkett and Klein, 2008] David Burkett and Dan Klein. Two languages are better than one (for syntactic parsing). In *Proceedings of the EMNLP*, 2008.

[Collins, 1999] Michael Collins. Head-driven statistical models for natural language parsing. In *Ph.D. Thesis*, 1999.

[Das and Petrov, 2011] Dipanjan Das and Slav Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL*, 2011.

[Ganchev *et al.*, 2009] Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. Dependency grammar induction via bitext projection constraints. In *Proceedings of the 47th ACL*, 2009.

[Headden III *et al.*, 2009] William P. Headden III, Mark Johnson, and David McClosky. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of HLT-NAACL*, 2009.

[Huang and Sagae, 2010] Liang Huang and Kenji Sagae. Dynamic programming for linear-time incremental parsing. In *Proceedings of ACL*, 2010.

[Huang *et al.*, 2009] Liang Huang, Wenbin Jiang, and Qun Liu. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of the EMNLP*, 2009.

[Hwa *et al.*, 2005] Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. In *Natural Language Engineering*, volume 11, pages 311–325, 2005.

[Jiang *et al.*, 2011] Wenbin Jiang, Qun Liu, and Yajuan Lü. Relaxed cross-lingual projection of constituent syntax. In *Proceedings of EMNLP*, 2011.

[Klein and Manning, 2004] Dan Klein and Christopher D. Manning. Corpus based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the ACL*, 2004.

[Koo and Collins, 2010] Terry Koo and Michael Collins. Efficient third-order dependency parsers. In *Proceedings of the ACL*, 2010.

[Marcus *et al.*, 1993] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. In *Computational Linguistics*, 1993.

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

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

[McDonald *et al.*, 2013] Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundagez, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Nuria Bertomeu Castello, and Jungmee Leez. Universal dependency annotation for multilingual parsing. In *Proceedings of ACL*, 2013.

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

[Nivre and Scholz, 2004] J. Nivre and M. Scholz. Deterministic dependency parsing of english text. In *Proceedings of the COLING*, 2004.

[Och, 2003] Franz Joseph Och. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*, pages 160–167, 2003.

[Parikh *et al.*, 2014] Ankur P. Parikh, Shay B. Cohen, and Eric P. Xing. Spectral unsupervised parsing with additive tree metrics. In *Proceedings of ACL*, 2014.

[Petrov *et al.*, 2006] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the ACL*, 2006.

[Sagae and Lavie, 2005] Kenji Sagae and Alon Lavie. A classifier-based parser with linear run-time complexity. In *Proceedings of IWPT*, 2005.

[Seginer, 2007] Yoav Seginer. Fast unsupervised incremental parsing. In *Proceedings of the ACL*, 2007.

[Smith and Eisner, 2009] David Smith and Jason Eisner. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*, 2009.

[Spitkovsky *et al.*, 2013] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In *Proceedings of EMNLP*, 2013.

[Tsuruoka and Tsujii, 2005] Yoshimasa Tsuruoka and Jun'ichi Tsujii. Chunk parsing revisited. In *Proceedings of IWPT*, 2005.

[Xue *et al.*, 2005] Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. The penn chinese treebank: Phrase structure annotation of a large corpus. In *Natural Language Engineering*, 2005.

[Yamada and Matsumoto, 2003] H Yamada and Y Matsumoto. Statistical dependency analysis using support vector machines. In *Proceedings of IWPT*, 2003.

[Zhang and Clark, 2008] Yue Zhang and Stephen Clark. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of EMNLP*, 2008.

[Zhu *et al.*, 2013] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. Fast and accurate shift-reduce constituent parsing. In *Proceedings of ACL*, 2013.