

# Bitext Dependency Parsing with Bilingual Subtree Constraints

Wenliang Chen, Jun'ichi Kazama and Kentaro Torisawa

Language Infrastructure Group, MASTAR Project

National Institute of Information and Communications Technology

3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto, Japan, 619-0289

{chenwl, kazama, torisawa}@nict.go.jp

## Abstract

This paper proposes a dependency parsing method that uses bilingual constraints to improve the accuracy of parsing bilingual texts (bitexts). In our method, a target-side tree fragment that corresponds to a source-side tree fragment is identified via word alignment and mapping rules that are automatically learned. Then it is verified by checking the subtree list that is collected from large scale automatically parsed data on the target side. Our method, thus, requires gold standard trees only on the source side of a bilingual corpus in the training phase, unlike the joint parsing model, which requires gold standard trees on the both sides. Compared to the re-ordering constraint model, which requires the same training data as ours, our method achieved higher accuracy because of richer bilingual constraints. Experiments on the translated portion of the Chinese Treebank show that our system outperforms monolingual parsers by 2.93 points for Chinese and 1.64 points for English.

## 1 Introduction

Parsing bilingual texts (bitexts) is crucial for training machine translation systems that rely on syntactic structures on either the source side or the target side, or the both (Ding and Palmer, 2005; Nakazawa et al., 2006). Bitexts could provide more information, which is useful in parsing, than a usual monolingual texts that can be called “bilingual constraints”, and we expect to obtain more accurate parsing results that can be effectively used in the training of MT systems. With this motivation, there are several studies aiming at highly

accurate bitext parsing (Smith and Smith, 2004; Burkett and Klein, 2008; Huang et al., 2009).

This paper proposes a dependency parsing method, which uses the bilingual constraints that we call *bilingual subtree constraints* and statistics concerning the constraints estimated from large unlabeled monolingual corpora. Basically, a (candidate) dependency subtree in a source-language sentence is mapped to a subtree in the corresponding target-language sentence by using word alignment and mapping rules that are automatically learned. The target subtree is verified by checking the subtree list that is collected from unlabeled sentences in the target language parsed by a usual monolingual parser. The result is used as additional features for the source side dependency parser. In this paper, our task is to improve the source side parser with the help of the translations on the target side.

Many researchers have investigated the use of bilingual constraints for parsing (Burkett and Klein, 2008; Zhao et al., 2009; Huang et al., 2009). For example, Burkett and Klein (2008) show that parsing with joint models on bitexts improves performance on either or both sides. However, their methods require that the training data have tree structures on both sides, which are hard to obtain. Our method only requires dependency annotation on the source side and is much simpler and faster. Huang et al. (2009) proposes a method, bilingual-constrained monolingual parsing, in which a source-language parser is extended to use the re-ordering of words between two sides' sentences as additional information. The input of their method is the source trees with their translation on the target side as ours, which is much easier to obtain than trees on both sides. However, their method does not use any tree structures on

the target side that might be useful for ambiguity resolution. Our method achieves much greater improvement because it uses the richer subtree constraints.

Our approach takes the same input as Huang et al. (2009) and exploits the subtree structure on the target side to provide the bilingual constraints. The subtrees are extracted from large-scale auto-parsed monolingual data on the target side. The main problem to be addressed is mapping words on the source side to the target subtree because there are many to many mappings and reordering problems that often occur in translation (Koehn et al., 2003). We use an automatic way for generating mapping rules to solve the problems. Based on the mapping rules, we design a set of features for parsing models. The basic idea is as follows: if the words form a subtree on one side, their corresponding words on the another side will also probably form a subtree.

Experiments on the translated portion of the Chinese Treebank (Xue et al., 2002; Bies et al., 2007) show that our system outperforms state-of-the-art monolingual parsers by 2.93 points for Chinese and 1.64 points for English. The results also show that our system provides higher accuracies than the parser of Huang et al. (2009).

The rest of the paper is organized as follows: Section 2 introduces the motivation of our idea. Section 3 introduces the background of dependency parsing. Section 4 proposes an approach of constructing bilingual subtree constraints. Section 5 explains the experimental results. Finally, in Section 6 we draw conclusions and discuss future work.

## 2 Motivation

In this section, we use an example to show the idea of using the bilingual subtree constraints to improve parsing performance.

Suppose that we have an input sentence pair as shown in Figure 1, where the source sentence is in English, the target is in Chinese, the dashed undirected links are word alignment links, and the directed links between words indicate that they have a (candidate) dependency relation.

In the English side, it is difficult for a parser to determine the head of word “with” because there is a PP-attachment problem. However, in Chinese it is unambiguous. Therefore, we can use the information on the Chinese side to help disambiguate.

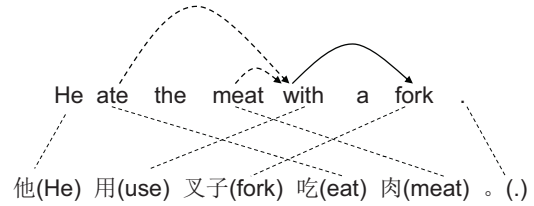


Figure 1: Example for disambiguation

tion.

There are two candidates “ate” and “meat” to be the head of “with” as the dashed directed links in Figure 1 show. By adding “fork”, we have two possible dependency relations, “meat-with-fork” and “ate-with-fork”, to be verified.

First, we check the possible relation of “meat”, “with”, and “fork”. We obtain their corresponding words “肉(meat)”, “用(use)”, and “叉子(fork)” in Chinese via the word alignment links. We verify that the corresponding words form a subtree by looking up a subtree list in Chinese (described in Section 4.1). But we can not find a subtree for them.

Next, we check the possible relation of “ate”, “with”, and “fork”. We obtain their corresponding words “吃(ate)”, “用(use)”, and “叉子(fork)”. Then we verify that the words form a subtree by looking up the subtree list. This time we can find the subtree as shown in Figure 2.

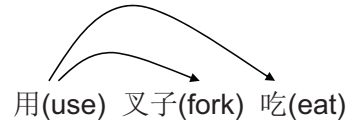


Figure 2: Example for a searched subtree

Finally, the parser may assign “ate” to be the head of “with” based on the verification results. This simple example shows how to use the subtree information on the target side.

## 3 Dependency parsing

For dependency parsing, there are two main types of parsing models (Nivre and McDonald, 2008; Nivre and Kubler, 2006): transition-based (Nivre, 2003; Yamada and Matsumoto, 2003) and graph-based (McDonald et al., 2005; Carreras, 2007). Our approach can be applied to both parsing models.

In this paper, we employ the graph-based MST parsing model proposed by McDonald and Pereira

(2006), which is an extension of the projective parsing algorithm of Eisner (1996). To use richer second-order information, we also implement parent-child-grandchild features (Carreras, 2007) in the MST parsing algorithm.

### 3.1 Parsing with monolingual features

Figure 3 shows an example of dependency parsing. In the graph-based parsing model, features are represented for all the possible relations on single edges (two words) or adjacent edges (three words). The parsing algorithm chooses the tree with the highest score in a bottom-up fashion.

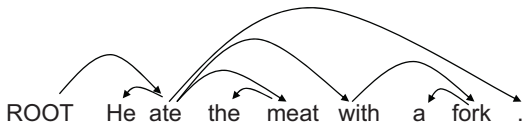


Figure 3: Example of dependency tree

In our systems, the monolingual features include the first- and second- order features presented in (McDonald et al., 2005; McDonald and Pereira, 2006) and the parent-child-grandchild features used in (Carreras, 2007). We call the parser with the monolingual features monolingual parser.

### 3.2 Parsing with bilingual features

In this paper, we parse source sentences with the help of their translations. A set of bilingual features are designed for the parsing model.

#### 3.2.1 Bilingual subtree features

We design bilingual subtree features, as described in Section 4, based on the constraints between the source subtrees and the target subtrees that are verified by the subtree list on the target side. The source subtrees are from the possible dependency relations.

#### 3.2.2 Bilingual reordering feature

Huang et al. (2009) propose features based on reordering between languages for a shift-reduce parser. They define the features based on word-alignment information to verify that the corresponding words form a contiguous span for resolving shift-reduce conflicts. We also implement similar features in our system.

## 4 Bilingual subtree constraints

In this section, we propose an approach that uses the bilingual subtree constraints to help parse source sentences that have translations on the target side.

We use large-scale auto-parsed data to obtain subtrees on the target side. Then we generate the mapping rules to map the source subtrees onto the extracted target subtrees. Finally, we design the bilingual subtree features based on the mapping rules for the parsing model. These features indicate the information of the constraints between bilingual subtrees, that are called bilingual subtree constraints.

### 4.1 Subtree extraction

Chen et al. (2009) propose a simple method to extract subtrees from large-scale monolingual data and use them as features to improve monolingual parsing. Following their method, we parse large unannotated data with a monolingual parser and obtain a set of subtrees ( $ST_t$ ) in the target language.

We encode the subtrees into string format that is expressed as  $st = w : hid(-w : hid)^+$ , where  $w$  refers to a word in the subtree and  $hid$  refers to the word ID of the word’s head ( $hid=0$  means that this word is the root of a subtree). Here, word ID refers to the ID (starting from 1) of a word in the subtree (words are ordered based on the positions of the original sentence). For example, “He” and “ate” have a left dependency arc in the sentence shown in Figure 3. The subtree is encoded as “He:2-ate:0”. There is also a parent-child-grandchild relation among “ate”, “with”, and “fork”. So the subtree is encoded as “ate:0-with:1-fork:2”. If a subtree contains two nodes, we call it a bigram-subtree. If a subtree contains three nodes, we call it a trigram-subtree.

From the dependency tree of Figure 3, we obtain the subtrees, as shown in Figure 4 and Figure 5. Figure 4 shows the extracted bigram-subtrees and Figure 5 shows the extracted trigram-subtrees. After extraction, we obtain a set of subtrees. We remove the subtrees occurring only once in the data. Following Chen et al. (2009), we also group the subtrees into different sets based on their frequencies.

<sup>1</sup>+ refers to matching the preceding element one or more times and is the same as a regular expression in Perl.

<pre>     ate    /  \   He   =&gt; He:1:2-ate:2:0 </pre>	<pre>     meat    /  \   the  =&gt; the:1:2-meat:2:0 </pre>
<pre>     ate    /  \   ate  =&gt; ate:1:0-meat:2:1   meat </pre>	<pre>     with    /  \   fork =&gt; with:1:0-fork:2:1 </pre>
<pre>     ate    /  \   with =&gt; ate:1:0-with:2:1 </pre>	<pre>     fork    /  \   a    =&gt; a:1:2-fork:2:0 </pre>

Figure 4: Examples of bigram-subtrees

<pre>     ate    /  \   meat with   /  \  ate  =&gt; ate:1:0-meat:2:1-with:3:1 He NULL </pre>	<pre>     ate    /  \   with   /  \  ate  =&gt; ate:1:0-with:2:1-:3:1 NULL meat </pre>
<pre>     ate    /  \   He  =&gt; He:1:3-NULL:2:3-ate:3:0 </pre>	<pre>     ate    /  \   NULL =&gt; ate:1:0-NULL:2:1-meat:3:1 </pre>
<pre>     the:1:3-NULL:2:3-meat:3:0 </pre>	<pre>     with:1:0-NULL:2:1-fork:3:1 </pre>
<pre>     a:1:3-NULL:2:3-fork:3:0 </pre>	

(a)

<pre>     ate:1:0-the:2:3-meat:3:1 </pre>	<pre>     ate:1:0-with:2:1-fork:3:2 </pre>
<pre>     with:1:0-a:2:3-fork:3:1 </pre>	<pre>     NULL:1:2-He:2:3-ate:3:0 </pre>
<pre>     He:1:3-NULL:2:1-ate:3:0 </pre>	<pre>     ate:1:0-meat:2:1-NULL:3:2 </pre>
<pre>     ate:1:0-NULL:2:3-with:3:1 </pre>	<pre>     with:1:0-fork:2:1-NULL:3:2 </pre>
<pre>     NULL:1:2-a:2:3-fork:3:0 </pre>	<pre>     a:1:3-NULL:2:1-fork:3:0 </pre>
<pre>     ate:1:0-NULL:2:3-:3:1 </pre>	<pre>     ate:1:0-:2:1-NULL:3:2 </pre>
<pre>     NULL:1:2-the:2:3-meat:3:0 </pre>	<pre>     the:1:3-NULL:2:1-meat:3:0 </pre>

(b)

Figure 5: Examples of trigram-subtrees

## 4.2 Mapping rules

To provide bilingual subtree constraints, we need to find the characteristics of subtree mapping for the two given languages. However, subtree mapping is not easy. There are two main problems: MtoN (words) mapping and reordering, which often occur in translation. MtoN (words) mapping means that a source subtree with M words is mapped onto a target subtree with N words. For example, 2to3 means that a source bigram-subtree is mapped onto a target trigram-subtree.

Due to the limitations of the parsing algorithm (McDonald and Pereira, 2006; Carreras, 2007), we only use bigram- and trigram-subtrees in our approach. We generate the mapping rules for the 2to2, 2to3, 3to3, and 3to2 cases. For trigram-subtrees, we only consider the parent-child-grandchild type. As for the use of other types of trigram-subtrees, we leave it for future work.

We first show the MtoN and reordering problems by using an example in Chinese-English translation. Then we propose a method to automatically generate mapping rules.

### 4.2.1 Reordering and MtoN mapping in translation

Both Chinese and English are classified as SVO languages because verbs precede objects in simple sentences. However, Chinese has many characteristics of such SOV languages as Japanese. The typical cases are listed below:

1) Prepositional phrases modifying a verb precede the verb. Figure 6 shows an example. In English the prepositional phrase “at the ceremony” follows the verb “said”, while its corresponding prepositional phrase “在(NULL) 仪式(ceremony) 上(at)” precedes the verb “说(say)” in Chinese.

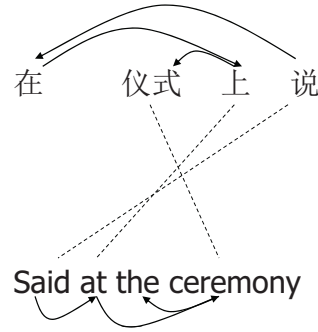


Figure 6: Example for prepositional phrases modifying a verb

2) Relative clauses precede head noun. Figure 7 shows an example. In Chinese the relative clause “今天(today) 签字(signed)” precedes the head noun “项目(project)”, while its corresponding clause “signed today” follows the head noun “projects” in English.

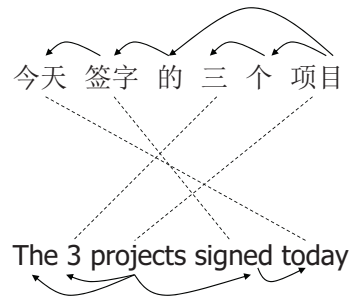


Figure 7: Example for relative clauses preceding the head noun

3) Genitive constructions precede head noun. For example, “汽车(car) 轮子(wheel)” can be translated as “the wheel of the car”.

4) Postposition in many constructions rather than prepositions. For example, “桌子(table) 上(on)” can be translated as “on the table”.

We can find the MtoN mapping problem occurring in the above cases. For example, in Figure 6, trigram-subtree “在(NULL):3-上(at):1-说(say):0” is mapped onto bigram-subtree “said:0-at:1”.

Since asking linguists to define the mapping rules is very expensive, we propose a simple method to easily obtain the mapping rules.

#### 4.2.2 Bilingual subtree mapping

To solve the mapping problems, we use a bilingual corpus, which includes sentence pairs, to automatically generate the mapping rules. First, the sentence pairs are parsed by monolingual parsers on both sides. Then we perform word alignment using a word-level aligner (Liang et al., 2006; DeNero and Klein, 2007). Figure 8 shows an example of a processed sentence pair that has tree structures on both sides and word alignment links.

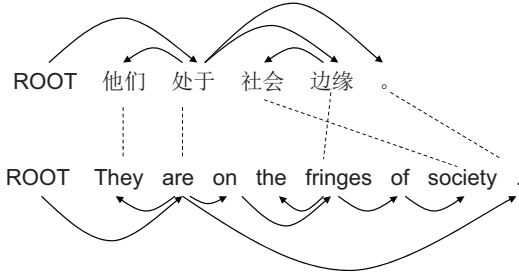


Figure 8: Example of auto-parsed bilingual sentence pair

From these sentence pairs, we obtain subtree pairs. First, we extract a subtree ( $st_s$ ) from a source sentence. Then through word alignment links, we obtain the corresponding words of the words of  $st_s$ . Because of the MtoN problem, some words lack of corresponding words in the target sentence. Here, our approach requires that at least two words of  $st_s$  have corresponding words and nouns and verbs need corresponding words. If not, it fails to find a subtree pair for  $st_s$ . If the corresponding words form a subtree ( $st_t$ ) in the target sentence,  $st_s$  and  $st_t$  are a subtree pair. We also keep the word alignment information in the target subtree. For example, we extract subtree “社会(society):2-边缘(fringe):0” on the Chinese side and get its corresponding subtree “fringes(W\_2):0-of:1-society(W\_1):2” on the English side, where W\_1 means that the target word is aligned to the first word of the source subtree, and W\_2 means that the target word is aligned to the second word of the source subtree. That is, we have a sub-

tree pair: “社会(society):2-边缘(fringe):0” and “fringe(W\_2):0-of:1-society(W\_1):2”.

The extracted subtree pairs indicate the translation characteristics between Chinese and English. For example, the pair “社会(society):2-边缘(fringe):0” and “fringes:0-of:1-society:2” is a case where “Genitive constructions precede/follow the head noun”.

#### 4.2.3 Generalized mapping rules

To increase the mapping coverage, we generalize the mapping rules from the extracted subtree pairs by using the following procedure. The rules are divided by “=>” into two parts: source (left) and target (right). The source part is from the source subtree and the target part is from the target subtree. For the source part, we replace *nouns* and *verbs* using their POS tags (coarse grained tags). For the target part, we use the word alignment information to represent the target words that have corresponding source words. For example, we have the subtree pair: “社会(society):2-边缘(fringe):0” and “fringes(W\_2):0-of:1-society(W\_1):2”, where “of” does not have a corresponding word, the POS tag of “社会(society)” is N, and the POS tag of “边缘(fringe)” is N. The source part of the rule becomes “N:2-N:0” and the target part becomes “W\_2:0-of:1-W\_1:2”.

Table 1 shows the top five mapping rules of all four types ordered by their frequencies, where W\_1 means that the target word is aligned to the first word of the source subtree, W\_2 means that the target word is aligned to the second word, and W\_3 means that the target word is aligned to the third word. We remove the rules that occur less than three times. Finally, we obtain 9,134 rules for 2to2, 5,335 for 2to3, 7,450 for 3to3, and 1,244 for 3to2 from our data. After experiments with different threshold settings on the development data sets, we use the top 20 rules for each type in our experiments.

The generalized mapping rules might generate incorrect target subtrees. However, as described in Section 4.3.1, the generated subtrees are verified by looking up list  $ST_t$  before they are used in the parsing models.

#### 4.3 Bilingual subtree features

Informally, if the words form a subtree on the source side, then the corresponding words on the target side will also probably form a subtree. For



#	rules	freq
<b>2to2 mapping</b>		
1	N:2 N:0 => W_1:2 W_2:0	92776
2	V:0 N:1 => W_1:0 W_2:1	62437
3	V:0 V:1 => W_1:0 W_2:1	49633
4	N:2 V:0 => W_1:2 W_2:0	43999
5	的:2 N:0 => W_2:0 W_1:2	25301
<b>2to3 mapping</b>		
1	N:2-N:0 => W_2:0-of:1-W_1:2	10361
2	V:0-N:1 => W_1:0-of:1-W_2:2	4521
3	V:0-N:1 => W_1:0-to:1-W_2:2	2917
4	N:2-V:0 => W_2:0-of:1-W_1:2	2578
5	N:2-N:0 => W_1:2-' :3-W_2:0	2316
<b>3to2 mapping</b>		
1	V:2-的/DEC:3-N:0 => W_1:0-W_3:1	873
2	V:2-的/DEC:3-N:0 => W_3:2-W_1:0	634
3	N:2-的/DEG:3-N:0 => W_1:0-W_3:1	319
4	N:2-的/DEG:3-N:0 => W_3:2-W_1:0	301
5	V:0-的/DEG:3-N:1 => W_3:0-W_1:1	247
<b>3to3 mapping</b>		
1	V:0-V:1-N:2 => W_1:0-W_2:1-W_3:2	9580
2	N:2-的/DEG:3-N:0 => W_3:0-W_2:1-W_1:2	7010
3	V:0-N:3-N:1 => W_1:0-W_2:3-W_3:1	5642
4	V:0-V:1-V:2 => W_1:0-W_2:1-W_3:2	4563
5	N:2-N:3-N:0 => W_1:2-W_2:3-W_3:0	3570

Table 1: Top five mapping rules of 2to3 and 3to2

example, in Figure 8, words “他们(they)” and “处于(be\_on)” form a subtree, which is mapped onto the words “they” and “are” on the target side. These two target words form a subtree. We now develop this idea as bilingual subtree features.

In the parsing process, we build relations for two or three words on the source side. The conditions of generating bilingual subtree features are that at least two of these source words must have corresponding words on the target side and nouns and verbs must have corresponding words.

At first, we have a possible dependency relation (represented as a source subtree) of words to be verified. Then we obtain the corresponding target subtree based on the mapping rules. Finally, we verify that the target subtree is included in  $ST_t$ . If yes, we activate a positive feature to encourage the dependency relation.

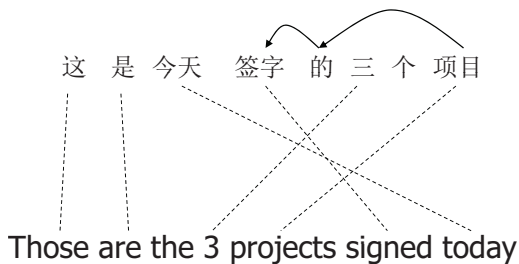


Figure 9: Example of features for parsing

We consider four types of features based on

2to2, 3to3, 3to2, and 2to3 mappings. In the 2to2, 3to3, and 3to2 cases, the target subtrees do not add new words. We represent features in a direct way. For the 2to3 case, we represent features using a different strategy.

### 4.3.1 Features for 2to2, 3to3, and 3to2

We design the features based on the mapping rules of 2to2, 3to3, and 3to2. For example, we design features for a 3to2 case from Figure 9. The possible relation to be verified forms source subtree “签字(signed)/VV:2-的(NULL)/DEC:3-项目(project)/NN:0” in which “项目(project)” is aligned to “projects” and “签字(signed)” is aligned to “signed” as shown in Figure 9. The procedure of generating the features is shown in Figure 10. We explain Steps (1), (2), (3), and (4) as follows:

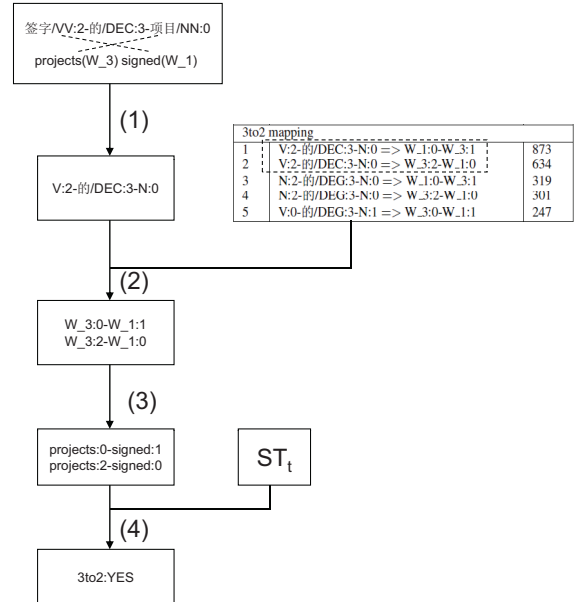


Figure 10: Example of feature generation for 3to2 case

(1) Generate source part from the source subtree. We obtain “V:2-的/DEC:3-N:0” from “签字(signed)/VV:2-的(NULL)/DEC:3-项目(project)/NN:0”.

(2) Obtain target parts based on the matched mapping rules, whose source parts equal “V:2-的/DEC:3-N:0”. The matched rules are “V:2-的/DEC:3-N:0 => W\_3:0-W\_1:1” and “V:2-的/DEC:3-N:0 => W\_3:2-W\_1:0”. Thus, we have two target parts “W\_3:0-W\_1:1” and “W\_3:2-W\_1:0”.

(3) Generate possible subtrees by consider-

ing the dependency relation indicated in the target parts. We generate a possible subtree “projects:0-signed:1” from the target part “W\_3:0-W\_1:1”, where “projects” is aligned to “项目(project)(W\_3)” and “signed” is aligned to “签字(signed)(W\_1)”. We also generate another possible subtree “projects:2-signed:0” from “W\_3:2-W\_1:0”.

(4) Verify that at least one of the generated possible subtrees is a target subtree, which is included in  $ST_t$ . If yes, we activate this feature. In the figure, “projects:0-signed:1” is a target subtree in  $ST_t$ . So we activate the feature “3to2:YES” to encourage dependency relations among “签字(signed)”, “的(NULL)”, and “项目(project)”.

### 4.3.2 Features for 2to3

In the 2to3 case, a new word is added on the target side. The first two steps are identical as those in the previous section. For example, a source part “N:2-N:0” is generated from “汽车(car)/NN:2-轮子(wheel)/NN:0”. Then we obtain target parts such as “W\_2:0-of/IN:1-W\_1:2”, “W\_2:0-in/IN:1-W\_1:2”, and so on, according to the matched mapping rules.

The third step is different. In the target parts, there is an added word. We first check if the added word is in the span of the corresponding words, which can be obtained through word alignment links. We can find that “of” is in the span “wheel of the car”, which is the span of the corresponding words of “汽车(car)/NN:2-轮子(wheel)/NN:0”. Then we choose the target part “W\_2:0-of/IN:1-W\_1:2” to generate a possible subtree. Finally, we verify that the subtree is a target subtree included in  $ST_t$ . If yes, we say feature “2to3:YES” to encourage a dependency relation between “汽车(car)” and “轮子(wheel)”.

## 4.4 Source subtree features

Chen et al. (2009) shows that the source subtree features ( $F_{src-st}$ ) significantly improve performance. The subtrees are obtained from the auto-parsed data on the source side. Then they are used to verify the possible dependency relations among source words.

In our approach, we also use the same source subtree features described in Chen et al. (2009). So the possible dependency relations are verified by the source and target subtrees. Combining two types of features together provides strong discrimination power. If both types of features are ac-

tive, building relations is very likely among source words. If both are inactive, this is a strong negative signal for their relations.

## 5 Experiments

All the bilingual data were taken from the translated portion of the Chinese Treebank (CTB) (Xue et al., 2002; Bies et al., 2007), articles 1-325 of CTB, which have English translations with gold-standard parse trees. We used the tool “Penn2Malt”<sup>2</sup> to convert the data into dependency structures. Following the study of Huang et al. (2009), we used the same split of this data: 1-270 for training, 301-325 for development, and 271-300 for test. Note that some sentence pairs were removed because they are not one-to-one aligned at the sentence level (Burkett and Klein, 2008; Huang et al., 2009). Word alignments were generated from the Berkeley Aligner (Liang et al., 2006; DeNero and Klein, 2007) trained on a bilingual corpus having approximately 0.8M sentence pairs. We removed notoriously bad links in {a, an, the} × {的(DE), 了(LE)} following the work of Huang et al. (2009).

For Chinese unannotated data, we used the XIN\_CMN portion of Chinese Gigaword Version 2.0 (LDC2009T14) (Huang, 2009), which has approximately 311 million words whose segmentation and POS tags are given. To avoid unfair comparison, we excluded the sentences of the CTB data from the Gigaword data. We discarded the annotations because there are differences in annotation policy between CTB and this corpus. We used the MMA system (Kruengkrai et al., 2009) trained on the training data to perform word segmentation and POS tagging and used the Baseline Parser to parse all the sentences in the data. For English unannotated data, we used the BLLIP corpus that contains about 43 million words of WSJ text. The POS tags were assigned by the MXPOST tagger trained on training data. Then we used the Baseline Parser to parse all the sentences in the data.

We reported the parser quality by the unlabeled attachment score (UAS), i.e., the percentage of tokens (excluding all punctuation tokens) with correct HEADs.

### 5.1 Main results

The results on the Chinese-source side are shown in Table 2, where “Baseline” refers to the systems

<sup>2</sup><http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html>

with monolingual features, “Baseline2” refers to adding the reordering features to the Baseline, “ $F_{BI}$ ” refers to adding all the bilingual subtree features to “Baseline2”, “ $F_{src-st}$ ” refers to the monolingual parsing systems with source subtree features, “Order-1” refers to the first-order models, and “Order-2” refers to the second-order models. The results showed that the reordering features yielded an improvement of 0.53 and 0.58 points (UAS) for the first- and second-order models respectively. Then we added four types of bilingual constraint features one by one to “Baseline2”. Note that the features based on 3to2 and 3to3 can not be applied to the first-order models, because they only consider single dependencies (bigram). That is, in the first model,  $F_{BI}$  only includes the features based on 2to2 and 2to3. The results showed that the systems performed better and better. In total, we obtained an absolute improvement of 0.88 points (UAS) for the first-order model and 1.36 points for the second-order model by adding all the bilingual subtree features. Finally, the system with all the features (OURS) outperformed the Baseline by an absolute improvement of 3.12 points for the first-order model and 2.93 points for the second-order model. The improvements of the final systems (OURS) were significant in McNemar’s Test ( $p < 10^{-4}$ ).

	Order-1	Order-2
Baseline	84.35	87.20
Baseline2	84.88	87.78
+2to2	85.08	88.07
+2to3	85.23	88.14
+3to3	–	88.29
+3to2	–	88.56
$F_{BI}$	85.23(+0.88)	88.56(+1.36)
$F_{src-st}$	86.54(+2.19)	89.49(+2.29)
OURS	87.47(+3.12)	90.13(+2.93)

Table 2: Dependency parsing results of Chinese-source case

We also conducted experiments on the English-source side. Table 3 shows the results, where abbreviations are the same as in Table 2. As in the Chinese experiments, the parsers with bilingual subtree features outperformed the Baselines. Finally, the systems (OURS) with all the features outperformed the Baselines by 1.30 points for the first-order model and 1.64 for the second-order model. The improvements of the final systems

(OURS) were significant in McNemar’s Test ( $p < 10^{-3}$ ).

	Order-1	Order-2
Baseline	86.41	87.37
Baseline2	86.86	87.66
+2to2	87.23	87.87
+2to3	87.35	87.96
+3to3	–	88.25
+3to2	–	88.37
$F_{BI}$	87.35(+0.94)	88.37(+1.00)
$F_{src-st}$	87.25(+0.84)	88.57(+1.20)
OURS	87.71(+1.30)	89.01(+1.64)

Table 3: Dependency parsing results of English-source case

## 5.2 Comparative results

Table 4 shows the performance of the system we compared, where Huang2009 refers to the result of Huang et al. (2009). The results showed that our system performed better than Huang2009. Compared with the approach of Huang et al. (2009), our approach used additional large-scale auto-parsed data. We did not compare our system with the joint model of Burkett and Klein (2008) because they reported the results on phrase structures.

	Chinese	English
Huang2009	86.3	87.5
Baseline	87.20	87.37
OURS	90.13	89.01

Table 4: Comparative results

## 6 Conclusion

We presented an approach using large automatically parsed monolingual data to provide bilingual subtree constraints to improve bitexts parsing. Our approach remains the efficiency of monolingual parsing and exploits the subtree structure on the target side. The experimental results show that the proposed approach is simple yet still provides significant improvements over the baselines in parsing accuracy. The results also show that our systems outperform the system of previous work on the same data.

There are many ways in which this research could be continued. First, we may attempt to apply the bilingual subtree constraints to transition-



based parsing models (Nivre, 2003; Yamada and Matsumoto, 2003). Here, we may design new features for the models. Second, we may apply the proposed method for other language pairs such as Japanese-English and Chinese-Japanese. Third, larger unannotated data can be used to improve the performance further.

## References

- Ann Bies, Martha Palmer, Justin Mott, and Colin Warner. 2007. English Chinese translation treebank v 1.0. In *LDC2007T02*.
- David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 877–886, Honolulu, Hawaii, October. Association for Computational Linguistics.
- X. Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 957–961.
- WL. Chen, J. Kazama, K. Uchimoto, and K. Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 570–579, Singapore, August. Association for Computational Linguistics.
- John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic, June. Association for Computational Linguistics.
- Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 541–548, Morristown, NJ, USA. Association for Computational Linguistics.
- J. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proc. of the 16th Intern. Conf. on Computational Linguistics (COLING)*, pages 340–345.
- Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1222–1231, Singapore, August. Association for Computational Linguistics.
- Chu-Ren Huang. 2009. Tagged Chinese Gigaword Version 2.0, LDC2009T14. *Linguistic Data Consortium*.
- P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*, page 54. Association for Computational Linguistics.
- Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. In *Proceedings of ACL-IJCNLP2009*, pages 513–521, Suntec, Singapore, August. Association for Computational Linguistics.
- Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA, June. Association for Computational Linguistics.
- R. McDonald and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of EACL2006*.
- R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proc. of ACL 2005*.
- T. Nakazawa, K. Yu, D. Kawahara, and S. Kurohashi. 2006. Example-based machine translation based on deeper nlp. In *Proceedings of IWSLT 2006*, pages 64–70, Kyoto, Japan.
- J. Nivre and S. Kubler. 2006. Dependency parsing: Tutorial at Coling-ACL 2006. In *CoLING-ACL*.
- J. Nivre and R. McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, June.
- J. Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of IWPT2003*, pages 149–160.
- David A. Smith and Noah A. Smith. 2004. Bilingual parsing with factored estimation: Using English to parse Korean. In *Proceedings of EMNLP*.
- Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated Chinese corpus. In *Coling*.
- H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT2003*, pages 195–206.
- Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. 2009. Cross language dependency parsing using a bilingual lexicon. In *Proceedings of ACL-IJCNLP2009*, pages 55–63, Suntec, Singapore, August. Association for Computational Linguistics.