

A Syntax-Driven Bracketing Model for Phrase-Based Translation

Deyi Xiong, Min Zhang, Aiti Aw and Haizhou Li

Human Language Technology

Institute for Infocomm Research

1 Fusionopolis Way, #21-01 Connexis, Singapore 138632

{dyxiong, mzhang, aaiti, hli}@i2r.a-star.edu.sg

Abstract

Syntactic analysis influences the way in which the source sentence is translated. Previous efforts add syntactic constraints to phrase-based translation by directly rewarding/punishing a hypothesis whenever it matches/violates source-side constituents. We present a new model that automatically learns syntactic constraints, including but not limited to constituent matching/violation, from training corpus. The model brackets a source phrase as to whether it satisfies the learnt syntactic constraints. The bracketed phrases are then translated as a whole unit by the decoder. Experimental results and analysis show that the new model outperforms other previous methods and achieves a substantial improvement over the baseline which is not syntactically informed.

1 Introduction

The phrase-based approach is widely adopted in statistical machine translation (SMT). It segments a source sentence into a sequence of phrases, then translates and reorder these phrases in the target. In such a process, original phrase-based decoding (Koehn et al., 2003) does not take advantage of any linguistic analysis, which, however, is broadly used in rule-based approaches. Since it is not linguistically motivated, original phrase-based decoding might produce ungrammatical or even wrong translations. Consider the following Chinese fragment with its parse tree:

Src: [把 [[7月 11日]_{NP} [设立 [为 [航海 节]_{NP}]_{PP}]_{VP}]_{IP}]_{VP}

Ref: established July 11 as Sailing Festival day

Output: [to/把 [[set up/设立 [for/为 navigation/航海]] on July 11/7月11日] knots/节]]

The output is generated from a phrase-based system which does not involve any syntactic analysis. Here we use “[]” (straight orientation) and “⟨⟩” (inverted orientation) to denote the common structure of the source fragment and its translation found by the decoder. We can observe that the decoder inadequately breaks up the second NP phrase and translates the two words “航海” and “节” separately. However, the parse tree of the source fragment constrains the phrase “航海 节” to be translated as a unit.

Without considering syntactic constraints from the parse tree, the decoder makes wrong decisions not only on phrase movement but also on the lexical selection for the multi-meaning word “节”¹. To avert such errors, the decoder can fully respect linguistic structures by only allowing syntactic constituent translations and reorderings. This, unfortunately, significantly jeopardizes performance (Koehn et al., 2003; Xiong et al., 2008) because by integrating syntactic constraint into decoding as a hard constraint, it simply prohibits any other useful non-syntactic translations which violate constituent boundaries.

To better leverage syntactic constraint yet still allow non-syntactic translations, Chiang (2005) introduces a count for each hypothesis and accumulates it whenever the hypothesis exactly matches syntactic boundaries on the source side. On the contrary, Marton and Resnik (2008) and Cherry (2008) accumulate a count whenever hypotheses violate constituent boundaries. These constituent matching/violation counts are used as a feature in the decoder’s log-linear model and their weights are tuned via minimal error rate training (MERT) (Och, 2003). In this way, syntactic constraint is integrated into decoding as a soft constraint to enable the decoder to reward hypotheses that respect syntactic analyses or to pe-

¹This word can be translated into “section”, “festival”, and “knot” in different contexts.

nalize hypotheses that violate syntactic structures.

Although experiments show that this constituent matching/violation counting feature achieves significant improvements on various language-pairs, one issue is that matching syntactic analysis can not always guarantee a good translation, and violating syntactic structure does not always induce a bad translation. Marton and Resnik (2008) find that some constituency types favor matching the source parse while others encourage violations. Therefore it is necessary to integrate more syntactic constraints into phrase translation, not just the constraint of constituent matching/violation.

The other issue is that during decoding we are more concerned with the question of phrase cohesion, i.e. whether the current phrase can be translated as a unit or not within particular syntactic contexts (Fox, 2002)², than that of constituent matching/violation. Phrase cohesion is one of the main reasons that we introduce syntactic constraints (Cherry, 2008). If a source phrase remains contiguous after translation, we refer this type of phrase **bracketable**, otherwise **unbracketable**. It is more desirable to translate a bracketable phrase than an unbracketable one.

In this paper, we propose a syntax-driven bracketing (SDB) model to predict whether a phrase (a sequence of contiguous words) is bracketable or not using rich syntactic constraints. We parse the source language sentences in the word-aligned training corpus. According to the word alignments, we define bracketable and unbracketable instances. For each of these instances, we automatically extract relevant syntactic features from the source parse tree as bracketing evidences. Then we tune the weights of these features using a maximum entropy (ME) trainer. In this way, we build two bracketing models: 1) a unary SDB model (UniSDB) which predicts whether an independent phrase is bracketable or not; and 2) a binary SDB model (BiSDB) which predicts whether two neighboring phrases are bracketable. Similar to previous methods, our SDB model is integrated into the decoder’s log-linear model as a feature so that we can inherit the idea of soft constraints.

In contrast to the constituent matching/violation counting (CMVC) (Chiang, 2005; Marton and Resnik, 2008; Cherry, 2008), our SDB model has

²Here we expand the definition of phrase to include both syntactic and non-syntactic phrases.

the following advantages

- The SDB model automatically learns syntactic constraints from training data while the CMVC uses manually defined syntactic constraints: constituency matching/violation. In our SDB model, each learned syntactic feature from bracketing instances can be considered as a syntactic constraint. Therefore we can use thousands of syntactic constraints to guide phrase translation.
- The SDB model maintains and protects the strength of the phrase-based approach in a better way than the CMVC does. It is able to reward non-syntactic translations by assigning an adequate probability to them if these translations are appropriate to particular syntactic contexts on the source side, rather than always punish them.

We test our SDB model against the baseline which does not use any syntactic constraints on Chinese-to-English translation. To compare with the CMVC, we also conduct experiments using (Marton and Resnik, 2008)’s XP+. The XP+ accumulates a count for each hypothesis whenever it violates the boundaries of a constituent with a label from {NP, VP, CP, IP, PP, ADVP, QP, LCP, DNP}. The XP+ is the best feature among all features that Marton and Resnik use for Chinese-to-English translation. Our experimental results display that our SDB model achieves a substantial improvement over the baseline and significantly outperforms XP+ according to the BLEU metric (Papineni et al., 2002). In addition, our analysis shows further evidences of the performance gain from a different perspective than that of BLEU.

The paper proceeds as follows. In section 2 we describe how to learn bracketing instances from a training corpus. In section 3 we elaborate the syntax-driven bracketing model, including feature generation and the integration of the SDB model into phrase-based SMT. In section 4 and 5, we present our experiments and analysis. And we finally conclude in section 6.

2 The Acquisition of Bracketing Instances

In this section, we formally define the bracketing instance, comprising two types namely binary bracketing instance and unary bracketing instance.

We present an algorithm to automatically extract these bracketing instances from word-aligned bilingual corpus where the source language sentences are parsed.

Let c and e be the source sentence and the target sentence, W be the word alignment between them, T be the parse tree of c . We define a **binary bracketing instance** as a tuple $\langle b, \tau(c_{i..j}), \tau(c_{j+1..k}), \tau(c_{i..k}) \rangle$ where $b \in \{\text{bracketable}, \text{unbracketable}\}$, $c_{i..j}$ and $c_{j+1..k}$ are two neighboring source phrases and $\tau(T, s)$ ($\tau(s)$ for short) is a subtree function which returns the minimal subtree covering the source sequence s from the source parse tree T . Note that $\tau(c_{i..k})$ includes both $\tau(c_{i..j})$ and $\tau(c_{j+1..k})$. For the two neighboring source phrases, the following conditions are satisfied:

$$\exists e_{u..v}, e_{p..q} \in e \text{ s.t.}$$

$$\forall (m, n) \in W, i \leq m \leq j \leftrightarrow u \leq n \leq v \quad (1)$$

$$\forall (m, n) \in W, j+1 \leq m \leq k \leftrightarrow p \leq n \leq q \quad (2)$$

The above (1) means that there exists a target phrase $e_{u..v}$ aligned to $c_{i..j}$ and (2) denotes a target phrase $e_{p..q}$ aligned to $c_{j+1..k}$. If $e_{u..v}$ and $e_{p..q}$ are neighboring to each other or all words between the two phrases are aligned to null, we set $b = \text{bracketable}$, otherwise $b = \text{unbracketable}$. From a binary bracketing instance, we derive a **unary bracketing instance** $\langle b, \tau(c_{i..k}) \rangle$, ignoring the subtrees $\tau(c_{i..j})$ and $\tau(c_{j+1..k})$.

Let n be the number of words of c . If we extract all potential bracketing instances, there will be $o(n^2)$ unary instances and $o(n^3)$ binary instances. To keep the number of bracketing instances tractable, we only record 4 representative bracketing instances for each index j : 1) the bracketable instance with the minimal $\tau(c_{i..k})$, 2) the bracketable instance with the maximal $\tau(c_{i..k})$, 3) the unbracketable instance with the minimal $\tau(c_{i..k})$, and 4) the unbracketable instance with the maximal $\tau(c_{i..k})$.

Figure 1 shows the algorithm to extract bracketing instances. Line 3-11 find all potential bracketing instances for each $(i, j, k) \in c$ but only keep 4 bracketing instances for each index j : two minimal and two maximal instances. This algorithm learns binary bracketing instances, from which we can derive unary bracketing instances.

```

1: Input: sentence pair  $(c, e)$ , the parse tree  $T$  of  $c$  and the
   word alignment  $W$  between  $c$  and  $e$ 
2:  $\mathfrak{R} := \emptyset$ 
3: for each  $(i, j, k) \in c$  do
4:   if There exist a target phrase  $e_{u..v}$  aligned to  $c_{i..j}$  and
      $e_{p..q}$  aligned to  $c_{j+1..k}$  then
5:     Get  $\tau(c_{i..j})$ ,  $\tau(c_{j+1..k})$ , and  $\tau(c_{i..k})$ 
6:     Determine  $b$  according to the relationship between
        $e_{u..v}$  and  $e_{p..q}$ 
7:     if  $\tau(c_{i..k})$  is currently maximal or minimal then
8:       Update bracketing instances for index  $j$ 
9:     end if
10:  end if
11: end for
12: for each  $j \in c$  do
13:    $\mathfrak{R} := \mathfrak{R} \cup \{\text{bracketing instances from } j\}$ 
14: end for
15: Output: bracketing instances  $\mathfrak{R}$ 

```

Figure 1: Bracketing Instances Extraction Algorithm.

3 The Syntax-Driven Bracketing Model

3.1 The Model

Our interest is to automatically detect phrase bracketing using rich contextual information. We consider this task as a binary-class classification problem: whether the current source phrase s is bracketable (b) within particular syntactic contexts ($\tau(s)$). If two neighboring sub-phrases s_1 and s_2 are given, we can use more inner syntactic contexts to complete this binary classification task.

We construct the syntax-driven bracketing model within the maximum entropy framework. A unary SDB model is defined as:

$$P_{UniSDB}(b|\tau(s), T) = \frac{\exp(\sum_i \theta_i h_i(b, \tau(s), T))}{\sum_b \exp(\sum_i \theta_i h_i(b, \tau(s), T))} \quad (3)$$

where $h_i \in \{0, 1\}$ is a binary feature function which we will describe in the next subsection, and θ_i is the weight of h_i . Similarly, a binary SDB model is defined as:

$$P_{BiSDB}(b|\tau(s_1), \tau(s_2), \tau(s), T) = \frac{\exp(\sum_i \theta_i h_i(b, \tau(s_1), \tau(s_2), \tau(s), T))}{\sum_b \exp(\sum_i \theta_i h_i(b, \tau(s_1), \tau(s_2), \tau(s), T))} \quad (4)$$

The most important advantage of ME-based SDB model is its capacity of incorporating more fine-grained contextual features besides the binary feature that detects constituent boundary violation or matching. By employing these features, we can investigate the value of various syntactic constraints in phrase translation.

new feature into the log-linear translation model: $P_{SDB}(b|T, \tau(\cdot))$. This feature is computed by the SDB model described in equation (3) or equation (4), which estimates a probability that a source span is to be translated as a unit within particular syntactic contexts. If a source span can be translated as a unit, the feature will give a higher probability even though this span violates boundaries of a constituent. Otherwise, a lower probability is given. Through this additional feature, we want the decoder to prefer hypotheses that translate source spans which can be translated as a unit, and avoids translating those which are discontinuous after translation. The weight of this new feature is tuned via MERT, which measures the extent to which this feature should be trusted.

In this paper, we implement the SDB model in a state-of-the-art phrase-based system which adapts a binary bracketing transduction grammar (BTG) (Wu, 1997) to phrase translation and reordering, described in (Xiong et al., 2006). Whenever a BTG merging rule ($s \rightarrow [s_1 s_2]$ or $s \rightarrow \langle s_1 s_2 \rangle$) is used, the SDB model gives a probability to the span s covered by the rule, which estimates the extent to which the span is bracketable. For the unary SDB model, we only consider the features from $\tau(s)$. For the binary SDB model, we use all features from $\tau(s_1)$, $\tau(s_2)$ and $\tau(s)$ since the binary SDB model is naturally suitable to the binary BTG rules.

The SDB model, however, is not only limited to phrase-based SMT using BTG rules. Since it is applied on a source span each time, any other hierarchical phrase-based or syntax-based system that translates source spans recursively or linearly, can adopt the SDB model.

4 Experiments

We carried out the MT experiments on Chinese-to-English translation, using (Xiong et al., 2006)’s system as our baseline system. We modified the baseline decoder to incorporate our SDB models as described in section 3.3. In order to compare with Marton and Resnik’s approach, we also adapted the baseline decoder to their XP+ feature.

4.1 Experimental Setup

In order to obtain syntactic trees for SDB models and XP+, we parsed source sentences using a lexicalized PCFG parser (Xiong et al., 2005). The parser was trained on the Penn Chinese Treebank

with an F1 score of 79.4%.

All translation models were trained on the FBIS corpus. We removed 15,250 sentences, for which the Chinese parser failed to produce syntactic parse trees. To obtain word-level alignments, we ran GIZA++ (Och and Ney, 2000) on the remaining corpus in both directions, and applied the “grow-diag-final” refinement rule (Koehn et al., 2005) to produce the final many-to-many word alignments. We built our four-gram language model using Xinhua section of the English Gigaword corpus (181.1M words) with the SRILM toolkit (Stolcke, 2002).

For the efficiency of MERT, we built our development set (580 sentences) using sentences not exceeding 50 characters from the NIST MT-02 set. We evaluated all models on the NIST MT-05 set using case-sensitive BLEU-4. Statistical significance in BLEU score differences was tested by paired bootstrap re-sampling (Koehn, 2004).

4.2 SDB Training

We extracted 6.55M bracketing instances from our training corpus using the algorithm shown in figure 1, which contains 4.67M bracketable instances and 1.89M unbracketable instances. From extracted bracketing instances we generated syntax-driven features, which include 73,480 rule features, 153,614 path features and 336 constituent boundary matching features. To tune weights of features, we ran the MaxEnt toolkit (Zhang, 2004) with iteration number being set to 100 and Gaussian prior to 1 to avoid overfitting.

4.3 Results

We ran the MERT module with our decoders to tune the feature weights. The values are shown in Table 1. The P_{SDB} receives the largest feature weight, 0.29 for UniSDB and 0.38 for BiSDB, indicating that the SDB models exert a nontrivial impact on decoder.

In Table 2, we present our results. Like (Marton and Resnik, 2008), we find that the XP+ feature obtains a significant improvement of 1.08 BLEU over the baseline. However, using all syntax-driven features described in section 3.2, our SDB models achieve larger improvements of up to 1.67 BLEU. The binary SDB (BiSDB) model statistically significantly outperforms Marton and Resnik’s XP+ by an absolute improvement of 0.59 (relatively 2%). It is also marginally better than the unary SDB model.

| System | Features | | | | | | | | | |
|----------|----------|----------|------------|------------|-------------|----------|------|-------|-------|-----------|
| | $P(c e)$ | $P(e c)$ | $P_w(c e)$ | $P_w(e c)$ | $P_{lm}(e)$ | $P_r(e)$ | Word | Phr. | XP+ | P_{SDB} |
| Baseline | 0.041 | 0.030 | 0.006 | 0.065 | 0.20 | 0.35 | 0.19 | -0.12 | — | — |
| XP+ | 0.002 | 0.049 | 0.046 | 0.044 | 0.17 | 0.29 | 0.16 | 0.12 | -0.12 | — |
| UniSDB | 0.023 | 0.051 | 0.055 | 0.012 | 0.21 | 0.20 | 0.12 | 0.04 | — | 0.29 |
| BiSDB | 0.016 | 0.032 | 0.027 | 0.013 | 0.13 | 0.23 | 0.08 | 0.09 | — | 0.38 |

Table 1: Feature weights obtained by MERT on the development set. The first 4 features are the phrase translation probabilities in both directions and the lexical translation probabilities in both directions. P_{lm} = language model; P_r = MaxEnt-based reordering model; Word = word bonus; Phr = phrase bonus.

| System | BLEU- n | n -gram Precision | | | | | | | |
|----------|-------------------|---------------------|------|------|------|-------|-------|-------|-------|
| | 4 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Baseline | 0.2612 | 0.71 | 0.36 | 0.18 | 0.10 | 0.054 | 0.030 | 0.016 | 0.009 |
| XP+ | 0.2720** | 0.72 | 0.37 | 0.19 | 0.11 | 0.060 | 0.035 | 0.021 | 0.012 |
| UniSDB | 0.2762**+ | 0.72 | 0.37 | 0.20 | 0.11 | 0.062 | 0.035 | 0.020 | 0.011 |
| BiSDB | 0.2779**++ | 0.72 | 0.37 | 0.20 | 0.11 | 0.065 | 0.038 | 0.022 | 0.014 |

Table 2: Results on the test set. **: significantly better than baseline ($p < 0.01$). + or ++: significantly better than Marton and Resnik’s XP+ ($p < 0.05$ or $p < 0.01$, respectively).

5 Analysis

In this section, we present analysis to perceive the influence mechanism of the SDB model on phrase translation by studying the effects of syntax-driven features and differences of 1-best translation outputs.

5.1 Effects of Syntax-Driven Features

We conducted further experiments using individual syntax-driven features and their combinations. Table 3 shows the results, from which we have the following key observations.

- The constituent boundary matching feature (CBMF) is a very important feature, which by itself achieves significant improvement over the baseline (up to 1.13 BLEU). Both our CBMF and Marton and Resnik’s XP+ feature focus on the relationship between a source phrase and a constituent. Their significant contribution to the improvement implies that this relationship is an important syntactic constraint for phrase translation.
- Adding more features, such as path feature and rule feature, achieves further improvements. This demonstrates the advantage of using more syntactic constraints in the SDB model, compared with Marton and Resnik’s XP+.

| Features | BLEU-4 | |
|----------------|-----------|--------------|
| | UniSDB | BiSDB |
| PF + RF | 0.2555 | 0.2644*@@ |
| PF | 0.2596 | 0.2671**@@ |
| CBMF | 0.2678** | 0.2725**@ |
| RF + CBMF | 0.2737** | 0.2780**++@@ |
| PF + CBMF | 0.2755**+ | 0.2782**++@- |
| RF + PF + CBMF | 0.2762**+ | 0.2779**++ |

Table 3: Results of different feature sets. * or **: significantly better than baseline ($p < 0.05$ or $p < 0.01$, respectively). + or ++: significantly better than XP+ ($p < 0.05$ or $p < 0.01$, respectively). @-: almost significantly better than its UniSDB counterpart ($p < 0.075$). @ or @@: significantly better than its UniSDB counterpart ($p < 0.05$ or $p < 0.01$, respectively).

- In most cases, the binary SDB is constantly significantly better than the unary SDB, suggesting that inner contexts are useful in predicting phrase bracketing.

5.2 Beyond BLEU

We want to further study the happenings after we integrate the constraint feature (our SDB model and Marton and Resnik’s XP+) into the log-linear translation model. In particular, we want to investigate: to what extent syntactic constraints change translation outputs? And in what direction the changes take place? Since BLEU is not sufficient

| System | CCM Rate (%) |
|----------|--------------|
| Baseline | 43.5 |
| XP+ | 74.5 |
| BiSDB | 72.4 |

Table 4: Consistent constituent matching rates reported on 1-best translation outputs.

to provide such insights, we introduce a new statistical metric which measures the proportion of syntactic constituents⁴ whose boundaries are consistently matched by decoder during translation. This proportion, which we call **consistent constituent matching (CCM) rate**, reflects the extent to which the translation output respects the source parse tree.

In order to calculate this rate, we output translation results as well as phrase alignments found by decoders. Then for each multi-branch constituent c_i^j spanning from i to j on the source side, we check the following conditions.

- If its boundaries i and j are aligned to phrase segmentation boundaries found by decoder.
- If all target phrases inside c_i^j 's target span⁵ are aligned to the source phrases within c_i^j and not to the phrases outside c_i^j .

If both conditions are satisfied, the constituent c_i^j is consistently matched by decoder.

Table 4 shows the consistent constituent matching rates. Without using any source-side syntactic information, the baseline obtains a low CCM rate of 43.53%, indicating that the baseline decoder violates the source parse tree more than it respects the source structure. The translation output described in section 1 is actually generated by the baseline decoder, where the second NP phrase boundaries are violated.

By integrating syntactic constraints into decoding, we can see that both Marton and Resnik's XP+ and our SDB model achieve a significantly higher constituent matching rate, suggesting that they are more likely to respect the source structure. The examples in Table 5 show that the decoder is able to generate better translations if it is

⁴We only consider multi-branch constituents.

⁵Given a phrase alignment $P = \{c_f^g \leftrightarrow e_p^a\}$, if the segmentation within c_i^j defined by P is $c_i^j = c_{i_1}^{j_1} \dots c_{i_k}^{j_k}$, and $c_{i_r}^{j_r} \leftrightarrow e_{u_r}^{v_r} \in P, 1 \leq r \leq k$, we define the **target span** of c_i^j as a pair where the first element is $\min(e_{u_1} \dots e_{u_k})$ and the second element is $\max(e_{v_1} \dots e_{v_k})$, similar to (Fox, 2002).

| System | CCM Rates (%) | | | | |
|--------|---------------|------|-------|-------|------|
| | <6 | 6-10 | 11-15 | 16-20 | >20 |
| XP+ | 75.2 | 70.9 | 71.0 | 76.2 | 82.2 |
| BiSDB | 69.3 | 74.7 | 74.2 | 80.0 | 85.6 |

Table 6: Consistent constituent matching rates for structures with different spans.

faithful to the source parse tree by using syntactic constraints.

We further conducted a deep comparison of translation outputs of BiSDB vs. XP+ with regard to constituent matching and violation. We found two significant differences that may explain why our BiSDB outperforms XP+. First, although the overall CCM rate of XP+ is higher than that of BiSDB, BiSDB obtains higher CCM rates for long-span structures than XP+ does, which are shown in Table 6. Generally speaking, violations of long-span constituents have a more negative impact on performance than short-span violations if these violations are toxic. This explains why BiSDB achieves relatively higher precision improvements for higher n -grams over XP+, as shown in Table 3.

Second, compared with XP+ that only punishes constituent boundary violations, our SDB model is able to encourage violations if these violations are done on bracketable phrases. We observed in many cases that by violating constituent boundaries BiSDB produces better translations than XP+ does, which on the contrary matches these boundaries. Still consider the example shown in section 1. The following translations are found by XP+ and BiSDB respectively.

XP+: [to/把 ⟨[set up/设立 [for the/为 [navigation/航海 section/节]]] on July 11/7月11日)]

BiSDB: [to/把 ⟨[[set up/设立 a/为] [marine/航海 festival/节]] on July 11/7月11日)]

XP+ here matches all constituent boundaries while BiSDB violates the PP constituent to translate the non-syntactic phrase “设立为”. Table 7 shows more examples. From these examples, we clearly see that appropriate violations are helpful and even necessary for generating better translations. By allowing appropriate violations to translate non-syntactic phrases according to particular syntactic contexts, our SDB model better inherits the strength of phrase-based approach than XP+.

| | |
|------------------|--|
| Src: | [[为 [印度洋 灾区 民众]NP]PP [奉献 [自己]NP [一份 爱心]NP]VP]VP |
| Ref: | show their loving hearts to people in the Indian Ocean disaster areas |
| Baseline: | <(love/爱心 [for the/为 <([people/民众 [to/奉献 [own/自己 a report/一份]])] <(in/灾区 the Indian Ocean/印度洋))> |
| XP+: | <([contribute/奉献 [its/自己 [part/一份 love/爱心]]) [for/为 <(the people/民众 <(in/灾区 the Indian Ocean/印度洋))>> |
| BiSDB: | <([[[contribute/奉献 its/自己] part/一份] love/爱心] [for/为 <(the people/民众 <(in/灾区 the Indian Ocean印度洋))>])> |
| Src: | [五角大厦 [已]ADVP [派遣 [[二十架]QP 飞机]NP [至 南亚]PP]VP]IP [,]PU [其中包括...]IP |
| Ref: | The Pentagon has dispatched 20 airplanes to South Asia, including... |
| Baseline: | [[The Pentagon/五角大厦 has sent/已派遣] [<u><[to/至 [[South Asia/南亚],] including/其中包括]] [20/二十 plane/架飞机]]]</u> |
| XP+: | [The Pentagon/五角大厦 [has/已 [sent/派遣 [[20/二十 planes/架飞机] [to/至 South Asia/南亚]]]] [/, [including/其中包括...]] |
| BiSDB: | [The Pentagon/五角大厦 [has sent/已派遣 [[20/二十 planes/架飞机] [to/至 South Asia/南亚]]] [/, [including/其中包括...]] |

Table 5: Translation examples showing that both XP+ and BiSDB produce better translations than the baseline, which inappropriately violates constituent boundaries (within underlined phrases).

| | |
|---------------|--|
| Src: | [[在 [[美国国务院 与 鲍尔]NP [短暂]ADJP [会谈]NP]NP 后]LCP]PP 表示]VP |
| Ref: | said after a brief discussion with Powell at the US State Department |
| XP+: | <[after/后 <([a brief/短暂 meeting/会谈] [with/与 Powell/鲍尔]) [in/在 the US State Department/美国国务院]) said/表示] |
| BiSDB: | <[said after/后 表示 <[a brief/短暂 meeting/会谈] <with Powell/与 鲍尔 [at/在 the State Department of the United States/美国国务院]]>> |
| Src: | [向 [[建立 [未来 民主 政治]NP]VP]IP]PP [迈出了 [关键性 的一步]NP]VP |
| Ref: | took a key step towards building future democratic politics |
| XP+: | <[a/了 [key/关键性 step/的一步]] <forward/迈出 [to/向 [a/建立 [future/未来 political democracy/民主政治]]]> |
| BiSDB: | <[made a/迈出了 [key/关键性 step/的一步]] [<u>towards establishing a/向 建立</u>] <democratic politics/民主政治 in the future/未来]> |

Table 7: Translation examples showing that BiSDB produces better translations than XP+ via appropriate violations of constituent boundaries (within double-underlined phrases).

6 Conclusion

In this paper, we presented a syntax-driven bracketing model that automatically learns bracketing knowledge from training corpus. With this knowledge, the model is able to predict whether source phrases can be translated together, regardless of matching or crossing syntactic constituents. We integrate this model into phrase-based SMT to increase its capacity of linguistically motivated translation without undermining its strengths. Experiments show that our model achieves substantial improvements over baseline and significantly outperforms (Marton and Resnik, 2008)’s XP+.

Compared with previous constituency feature, our SDB model is capable of incorporating more syntactic constraints, and rewarding necessary violations of the source parse tree. Marton and Resnik (2008) find that their constituent constraints are sensitive to language pairs. In the future work, we will use other language pairs to test

our models so that we could know whether our method is language-independent.

References

- Colin Cherry. 2008. Cohesive Phrase-based Decoding for Statistical Machine Translation. In *Proceedings of ACL*.
- David Chiang. 2005. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of ACL*, pages 263–270.
- David Chiang, Yuval Marton and Philip Resnik. 2008. Online Large-Margin Training of Syntactic and Structural Translation Features. In *Proceedings of EMNLP*.
- Heidi J. Fox. 2002. Phrasal Cohesion and Statistical Machine Translation. In *Proceedings of EMNLP*, pages 304–311.
- Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of HLT-NAACL*.

- Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP*.
- Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *International Workshop on Spoken Language Translation*.
- Yajuan Lü, Sheng Li, Tiezhun Zhao and Muyun Yang. 2002. Learning Chinese Bracketing Knowledge Based on a Bilingual Language Model. In *Proceedings of COLING*.
- Yuval Marton and Philip Resnik. 2008. Soft Syntactic Constraints for Hierarchical Phrase-Based Translation. In *Proceedings of ACL*.
- Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of ACL 2000*.
- Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL 2003*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatically Evaluation of Machine Translation. In *Proceedings of ACL*.
- Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of International Conference on Spoken Language Processing*, volume 2, pages 901-904.
- Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377-403.
- Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, Yueliang Qian. 2005. Parsing the Penn Chinese Treebank with Semantic Knowledge. In *Proceedings of IJCNLP*, Jeju Island, Korea.
- Deyi Xiong, Qun Liu and Shouxun Lin. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *Proceedings of ACL-COLING 2006*.
- Deyi Xiong, Min Zhang, Aiti Aw, and Haizhou Li. 2008. Linguistically Annotated BTG for Statistical Machine Translation. In *Proceedings of COLING 2008*.
- Le Zhang. 2004. Maximum Entropy Modeling Toolkit for Python and C++. Available at http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.