

# Machine Translation by Modeling Predicate-Argument Structure Transformation

Feifei ZHAI, Jiajun ZHANG, Yu ZHOU and Chengqing ZONG

National Laboratory of Pattern Recognition, Institute of Automation,  
Chinese Academy of Sciences, Beijing, China

{ffzhai, jjzhang, yzhou, cqzong}@nlpr.ia.ac.cn

## ABSTRACT

Machine translation aims to generate a target sentence that is semantically equivalent to the source sentence. However, most of current statistical machine translation models do not model the semantics of sentences. In this paper, we propose a novel translation framework based on predicate-argument structure (PAS) for its capacity on grasping the semantics and skeleton structure of sentences. By using PAS, the framework effectively models both semantics of languages and global reordering for translation. In the framework, we divide the translation process into 3 steps: (1) *PAS acquisition*: perform semantic role labeling (SRL) on the input sentences to acquire source-side PASs; (2) *Transformation*: convert source-side PASs to their target counterparts by predicate-aware PAS transformation rules; (3) *Translation*: first translate the predicate and arguments of PAS and then adopt a CKY-style decoding algorithm to translate the entire PAS. Experimental results show that our PAS-based translation framework significantly improves the translation performance.

---

KEYWORDS: Predicate-argument structure; Semantic role labeling; PAS transformation; PAS-based translation

---

## 1 Introduction

Statistical machine translation (SMT) has made significant progress from word-based models (Brown et al., 1993) to phrase-based models (Koehn et al., 2003; Och and Ney, 2004) and syntax-based models (Galley et al., 2006; Liu et al., 2006; Marcu et al., 2006) over the past decades. However, the existing SMT models are always criticized for not modeling the semantics of languages. Furthermore, reordering is always one of the most difficult and important research problems in SMT. However, although current translation models are much good at local reordering<sup>1</sup>, most of them are weak to cope with global reordering<sup>2</sup>. The two weaknesses restrict current translation models a lot, which urges us to seek a new translation framework to model both the semantics of languages and global reordering.

Formally, predicate-argument structure (PAS) is a structure that depicts the relationship between a predicate and its associated arguments, and it always indicates the semantic frame and skeleton structure of a sentence. From the characteristics of PAS, we can see that it provides not only a good semantic representation for modeling semantics, but also a skeleton structure for global reordering. Moreover, Fung et al. (2006) and Wu and Fung (2009b) have shown that PASs of the both sides are more consistent with each other than syntax structures. Considering current syntax-based translation models are always impaired by cross-lingual structure divergence (Eisner, 2003; Zhang et al., 2010), PAS will be a better alternative for building translation models.

Therefore, in this paper, aiming at building a PAS-based translation framework, we propose a novel translation method based on PAS transformation. FIGURE 1 is an overview of our method. Specifically, we divide the entire translation process into 3 steps:

- (1) *PAS acquisition*: perform semantic role labeling (SRL) on the input sentences to achieve their PASs, i.e., *source-side PASs*.
- (2) *Transformation*: convert source-side PASs to target-side-like PASs by predicate-aware PAS transformation rules, which are extracted from the result of bilingual semantic role labeling (Zhuang and Zong, 2010b). Here, *target-side-like PAS* denotes a list of general non-terminals in target language order, where a non-terminal aligns to a source element. Henceforward, we use source elements to denote the predicate and arguments of source-side PAS (similarly for target elements).
- (3) *Translation*: just as FIGURE 1 shows, this step is further divided into two parts: (a) *element translation* is to translate each source element respectively; (b) *translation by global reordering* is to combine the translation candidates of source elements to translate the entire PAS based on the target-side-like PAS.

This method performs translation based on the PASs of sentences. In the transformation step, we model the source-side PAS by PAS transformation rules and convert it to target-side-like PAS. This means that we transform the skeleton structure of source sentence into the skeleton structure of target language. Obviously, this transformation process relates both sides on the skeleton level and would be potential to handle the global reordering problem.

---

<sup>1</sup> In this paper, global reordering refers to perform reordering based on the entire sentence structure. The other reordering operations are actually all local ones, even for the long-distance reordering without considering the global sentence structure.

<sup>2</sup> Only syntax-based models have tried to model global reordering. However, it needs large translation rules to take the entire sentence structure into account. This requirement always leads to a severe sparsity problem for translation. Therefore, the global reordering problem is not well addressed in these models.

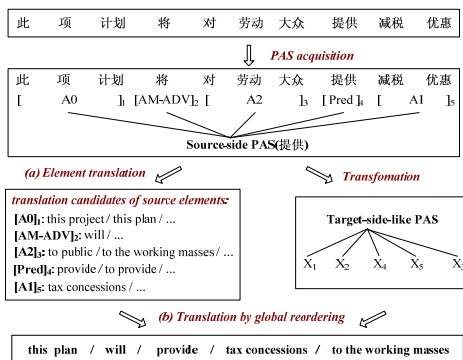


FIGURE 1 Three steps of our PAS-based translation framework: (1) *PAS acquisition*; (2) *Transformation*; (3) *Translation*. In the figure, the same subscript denotes the one-to-one alignment between source elements and non-terminals of target-side-like PAS.

Intuitively, when a human interpreter translates a sentence, he/she segments the sentence according to his/her understanding and translates each part respectively and then he/she translates the entire sentence by combining the partial translation of all parts. From this sense, the translation process of our PAS-based translation framework is similar to human translation to some extent. We believe that this work is a big step towards semantics-based machine translation.

Remainder of the paper is structured as follows. Section 2 elaborates the automatic process of extracting the predicate-aware PAS transformation rules. Section 3 details the translation process of our method. Section 4 describes how to decode the whole sentence with our method. In section 5, we evaluate the effectiveness of our method and in section 6, we introduce the related work. Finally, we end with the conclusion and perspectives.

## 2 PAS Transformation Rule Extraction

In this section, we introduce the method of bilingual semantic role labeling (SRL) and present how to extract PAS transformation rules based on the bilingual SRL result.

### 2.1 Bilingual Semantic Role Labeling

Bilingual SRL is to perform SRL on bitext simultaneously. In order to do this, (Zhuang and Zong, 2010b) proposed a method to infer bilingual semantic roles jointly. At first, they looked for aligned bilingual predicates and generated multiple monolingual SRL results by monolingual SRL systems. Then they adopted an integer linear programming method to find the best bilingual SRL result. They not only achieved the start-of-the-art monolingual SRL performance to date, but acquired the mapping between bilingual arguments. Thus, we follow their work to achieve bilingual SRL results for our training set. FIGURE 2(a) shows an example of bilingual SRL.

### 2.2 Rule Extraction

With the bilingual SRL result in FIGURE 2(a), we can easily generate an *exact transformation rule*, of which the left and right side is the PASs on the two sides, just as FIGURE 2(b) shows. Using the

rule, we can project the translation candidates of source elements to their aligned target elements and then translate the entire PAS by combining these candidates.

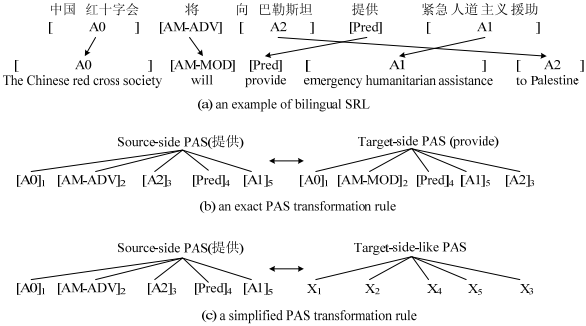


FIGURE 2 – An example of bilingual SRL and the corresponding PAS transformation rules: In (b) and (c), the same subscript at the source and target side denotes the aligned elements in PASs.

Obviously, semantic roles of target elements are not used in the above translation process<sup>3</sup>. Therefore, we can simplify the exact transformation rule by substituting target elements’ semantic roles with general non-terminals. We call the achieved target-side PAS as *target-side-like PAS* and name the rule as *simplified transformation rule*, just like the rule in FIGURE 2(c). Basically, a simplified transformation rule  $r$  is a triple  $\langle Pred, SP, TP \rangle$ :

- $Pred$  is the specific source-side predicate where rule  $r$  is extracted.
- $SP$  denotes the source-side PAS, which is a list of source elements in source language order.
- $TP$  is the target-side-like PAS, i.e., a list of general non-terminals in target language order.

For example, the rule in FIGURE 2(c) is a tripe where  $Pred$  is Chinese verb “提供”,  $SP$  is the source element list  $\langle [A0]_1 [AM-ADV]_2 [A2]_3 [Pred]_4 [A1]_5 \rangle$ , and  $TP$  is the list of non-terminals  $\langle X_1 X_2 X_4 X_5 X_3 \rangle$ . The same subscript in  $SP$  and  $TP$  refer to the one-to-one mapping between a source element and a target non-terminal. Obviously, the transformation rule can easily grasp the interrelation of bilingual PASs. Note that the target predicate “provide” in FIGURE 2(b) is ignored because its counterpart predicate “提供” will be translated by the element  $[Pred]$  in  $SP$ .

Virtually, in order to project the translation candidates of source elements to target-side-like PAS, we require that a source argument only aligns to a target argument. However, the result of bilingual SRL usually does not satisfy this requirement. There exist many unaligned source arguments, and sometimes a source argument might align to more than one target argument.

To resolve this problem, we refine the bilingual SRL result via word alignment. We focus on source arguments and refine the corresponding target arguments. For the unaligned source arguments, we look for their target spans via word alignment. If the source argument and its target span are consistent with word alignment<sup>4</sup>, and its target span does not overlap with the

<sup>3</sup> Semantic roles of target elements can be used to evaluate the quality of translation candidates. Here we do not consider this point and we take it as our future work.

<sup>4</sup> Two spans are consistent with word alignment means that words in source span only align to words in target span via word alignment, and vice versa.

target span of other source arguments, we take the target span as a virtual target argument for rule extraction. Otherwise, we ignore the source argument.

Towards the source argument aligning to more than one target argument, we check the minimal continuous target span covering all its aligned target arguments. If the span does not overlap with other target arguments, we also take the span as a virtual argument for rule extraction. Otherwise, we discard the source argument. In addition, for the predicate whose multiple arguments align to one or more target arguments (many-to-one/many case), we do not extract rules from that predicate. According to our final statistics, only 6.9% of the aligned predicate pairs are discarded.

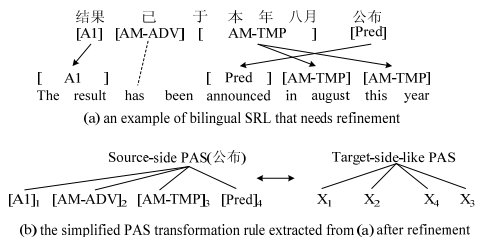


FIGURE 3 – An example for refining the bilingual SRL result.

For example, in FIGURE 3, although the source argument [AM-ADV] is unaligned, we align it to target word “has” via word alignment. For source argument [AM-TMP], the minimal span that covers the two target argument [AM-TMP]s does not overlap with other target arguments. We take that span as a big virtual argument for rule extraction. At last, we extract the simplified transformation rule in FIGURE 3(b).

Finally, the transformation rules are organized into a Trie structure. In order to store a rule, we use the rule’s *Pred* and *SP* as the key, and *TP* as the value of Trie node. Henceforward, we utilize *TRTrie* to denote the Trie structure encoding all the transformation rules.

### 2.3 Rule Extension

Basically, some modifier arguments<sup>5</sup> are actually not necessary for the skeleton of sentences. For example, source argument [AM-TMP] in FIGURE 3(a) is a modifier. If we ignore it and its target counterpart, the remaining PAS is still reasonable. Therefore, we extend the PAS transformation rules based on this insight. For a specific PAS transformation rule, we traverse all its modifiers and discard each one in turn, and meanwhile, construct a simplified transformation rule with the remaining arguments of the PAS. For instance, if we ignore the source argument [AM-TMP] in FIGURE 3(a), we can get a simplified transformation rule where *Pred* is verb “公布”, *SP* is the source element list  $\langle [AI]_1 [AM-ADV]_2 [Pred]_3 \rangle$ , and *TP* is  $\langle X_1 X_2 X_3 \rangle$ .

### 2.4 Rule Probabilities

To distinguish different transformation rules during decoding, we design two probabilities for each transformation rule: *predicate-conditioned rule probability*  $p_{pred}(r)$  and *source-PAS-conditioned rule probability*  $p_{sp}(r)$ :

<sup>5</sup> The argument that utilizes AM as its prefix.

$$P_{pred}(r) = \frac{c(r)}{\sum_{r': Pred(r')=Pred(r)} c(r')}$$

$$P_{sp}(r) = \frac{c(TSP(r))}{\sum_{r': SP(r')=SP(r)} c(TSP(r'))}$$

In the two formulas,  $Pred(r)$  and  $SP(r)$  denote  $Pred$  and  $SP$  of rule  $r$  respectively.  $TSP(r)$  refers to the combination of rule  $r$ 's  $SP$  and  $TP$ .  $c(r)$  is the count of rule  $r$  (similarly for  $c(TSP(r))$ ). The two probabilities will serve as features for decoding. Generally, the first feature is mainly used to evaluate which transformation rule is more possible for the specific source predicate. The second feature is used to evaluate which  $TP$  is more appropriate for the specific  $SP$ <sup>6</sup>. The two features indicate the distribution of bilingual PASs from two different angles, which will be helpful for the decoder to choose effective PAS transformation rules.

### 3 PAS-based Translation Framework

In the *PAS acquisition* step, we perform SRL on each test sentence with a monolingual SRL system. To alleviate the negative impact of SRL errors, we use multiple SRL results. We provide the monolingual SRL system with 3-best parse trees of Berkeley parser (Petrov and Klein, 2007), 1-best parse tree of Bikel parser (Bikel, 2004) and Stanford parser (Klein and Manning, 2003). FIGURE 4(a) shows an example of multiple SRL results. In the *transformation* step, we match the multiple SRL results with PAS transformation rules and convert them to target-side-like PASs. Then in the *translation* step, we decode the PAS based on these target-side-like PASs.

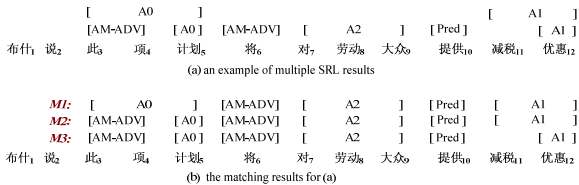


FIGURE 4 – Multiple SRL results and the final matching result of the example sentence.

#### 3.1 PAS Transformation

In this section, we describe how to match the multiple SRL results with PAS transformation rules and transform them to target-side-like PASs. We design *Algorithm 1* to achieve our purpose. First, we look for the predicate in  $TRTrie$  and get the matching Trie node  $P\_N$ . With this node, we continuously match the elements of PAS in order, and meanwhile, expand along  $TRTrie$ . Finally, we achieve all possible PASs that can match transformation rules. We only preserve the ones covering the largest number of source words or elements. We believe that only the PAS satisfying one of the two conditions is possible to stand for the real skeleton of a sentence and capture a good global reordering operation. For example, FIGURE 4(b) shows the matching result of FIGURE 4(a). The result M1 in FIGURE 4(b) covers the largest number of source words, and M3 carries the largest number of elements, and moreover, M2 satisfies the two conditions. After that,

<sup>6</sup> Actually, this feature should base on the entire rule  $r$ , rather than  $TSP(r)$ . However, this leads to severe data sparseness for rules. Therefore, we pursue the general rules and ignore the predicate here.

we can get target-side-like PASs from the transformation rules. Algorithm 1’s complexity is exponential, but its speed is fast in practice because a predicate only carries very few arguments.

---

**Algorithm 1: PAS Transformation Rule Matching**

---

Input: predicate  $P$ , a list  $L$  including all the source elements of  $P$ , and  $TRTrie$

Output: a list  $TPL$  preserving all the achieved target-side-like PASs

```

1: function Matching ( $P, L, TRTrie$ ):
2:   sort  $L$  first by the element’s start position and then by its length from small to large
3:   find  $P$  in  $TRTrie$  and get the Trie node  $P\_N$ , if not find  $P$ , return      < match the predicate first
4:   for  $c\_arg$  in  $L$  do:                                                    < consider all elements in turn
5:     for  $p\_arg$  that is before  $c\_arg$  in  $L$  do:                            < check all partial matching PASs
6:       if  $p\_arg$  does not overlap with  $c\_arg$ :
7:         for Trie node  $t\_n$  in  $p\_arg$  do:
8:           if  $c\_arg$  in descendents of  $t\_n$ , then store that node into  $c\_arg$  < expand along  $TRTrie$ 
9:           find  $c\_arg$  in descendents of  $P\_N$ , if find, store that node into  $c\_arg$  < PAS might begin with any element
10:        check all Trie nodes stored in  $L$ ’s elements, consider the rules covering the largest number of arguments or
            source words, and save  $TPL$ s of these rules into  $TPL$           < store the target-side-like PAS
11:   return  $TPL$ 

```

---

We use *matching score* to evaluate the matching PASs. For a PAS  $A_{m1}, \dots, A_{mm}$ , such as  $\langle [A0][AM-ADV][A2][Pred][A1] \rangle$  (the matching result M1 in FIGURE 4(b)), its matching score is:

$$P_{ms}(A_{m1}, \dots, A_{mm}) = \frac{\prod_j p(A_{mj} | S, pred)}{\sum_{m'} \prod_j p(A_{m'j} | S, pred)}$$

where  $S$  and  $pred$  denote the test sentence and the predicate respectively.  $p(A_{mj} | S, pred)$  denotes the probability that the SRL system assigns to element  $A_{mj}$ <sup>7</sup>. Additionally, the denominator sums the score of all matching PASs. This matching score will serve as a feature in the final decoder. It is mainly used to reward the good skeleton structure of sentences.

### 3.2 Gap Word Attachment

In a matching PAS, adjacent source elements might be separated by *gap words* in the sentence. For example, in the matching result M3 of FIGURE 4(b), [Pred] and [A1] are separated by a gap word “减税”. For the PAS whose elements are separated by gap words, we cannot translate it only based on the target-side-like PAS because it is not continuous. Therefore, to address this problem, we attach the gap words to their neighbouring left or right elements via parse tree. We look for the lowest common ancestor nodes of the gap word and its left or right neighbouring elements respectively. We compare these two ancestor nodes and attach the gap word to the element whose corresponding ancestor node is lower in the parse tree. For example in FIGURE 5, the common ancestor node of word “减税” and [A1] is node  $NP_{11,12}$ , while it is node  $VP_{10,12}$  for [Pred]. Hence, we attach word “减税” to [A1] and transform the  $PAS_1$  to  $PAS_2$  in FIGURE 5.

In practice, it is common that the neighboring left and right elements get the same ancestor node. This is because a father node can dominate many children nodes in parse trees. To address this problem, we employ the head binarization method (Wang et al., 2007) to binarize the parse trees.

---

<sup>7</sup> We average the five probabilities given by the 5 parse trees as this probability.

We make the final attachment decision by voting with the abovementioned five parse trees. After attachment, some PASs may be identical to each other, such as the matching result M2 and M3 of FIGURE 4(b). We only retain the one whose matching score is larger.

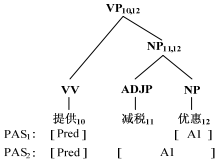


FIGURE 5 – An example of gap word attachment using parse tree.

### 3.3 PAS Translation

In the *translation* step, we translate each source element by a traditional translation method. Then we combine these candidates to translate the entire PAS based on the target-side-like PAS, just as FIGURE 1 shows. Intuitively, the combination can be operated directly by cube pruning (Chiang, 2007). However, since the source elements are translated independently and many source elements’ spans are very short, numerous phrase translation rules are ignored during translation. This fact leads to a narrow decoding space and poor translation accuracy. To alleviate this problem, we design a CKY-style decoding algorithm for each target-side-like PAS.

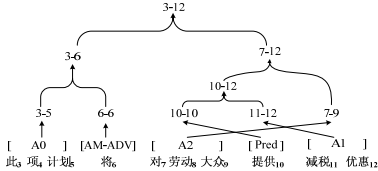


FIGURE 6 – An example of our CKY-style decoding algorithm for target-side-like PAS. In this example, only one path is generated for the final span 3-12. In practice, there can be many paths.

In the CKY-style decoding algorithm, we organize the source elements in target language order based on the target-side-like PAS. For example, in FIGURE 6, we use the rule in FIGURE 2(c) and create the span list [3,5], [6,6], [10,10], [11,12], [7,9]. Then we combine these spans in a bottom-up manner, just like traditional CKY algorithm works. The difference is that we only check all the possible combinations of small spans to form big spans, rather than checking all the split points of a big span. Moreover, if the adjacent spans are not adjacent at the source side, we do not combine them. For instance, in FIGURE 6, span [6,6] and [10,10] are adjacent in target order, but they are not adjacent at the source side. In addition, the translation candidates of newly generated spans, such as span [3,6], come from two parts: combining the translation candidates of its two sub-spans by cube pruning, or using phrase translation rules. These combined spans help to enlarge the search space a lot and yield a good translation performance.

Basically, only when the target-side-like PAS can be binarized, our decoding algorithm can be implemented. According to our statistics, almost all the target-side-like PASs can be binarized. We will detail the statistics in sub-section 5.2. If a target-side-like PAS cannot be binarized, we combine the partial translations of its elements by cube pruning straightforwardly.



## 4 Decoding with PAS-based Translation Framework

Formally, PAS represents the main structure of a sentence. However, sometimes the sentence cannot be fully covered by a PAS, especially when there are several predicates in the sentence. In order to translate the whole sentence, we design a decoding algorithm in terms of our PAS-based translation framework. The algorithm we adopted here follows the CKY-style framework.

In the decoder, we organize the search space of translation candidates into a hypergraph. For the span covered by PAS (named as *PAS span*), we use a multiple-branch hyperedge to connect that span to the PAS's elements. For the span not covered by PAS (named as *non-PAS span*), we consider all the binary segmentations of that span and use binary hyperedges to link them, just as FIGURE 7 shows. As a realistic example, FIGURE 8(a) shows a sentence and the PAS of its predicate “说(say)”. The PASs of another predicate “提供(provide)” in the sentence are shown in FIGURE 4(b). The final decoding hypergraph is shown in FIGURE 8(b).

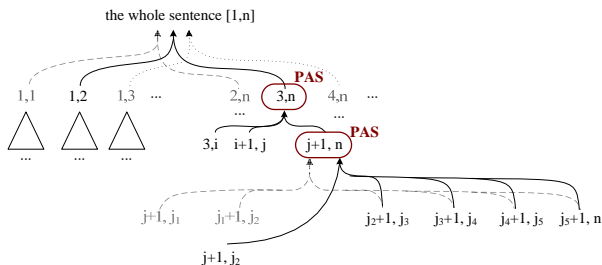


FIGURE 7 – An illustration of the decoding hypergraph. In the FIGURE,  $n$  refers to the length of sentence. Span  $[3,n]$  and  $[j+1,n]$  denote PAS spans and their descendent spans are all spans of elements in PAS.

After the hypergraph is constructed, we fill the spans with translation candidates in a bottom-up manner. When we encounter a PAS span, the algorithm described in sub-section 3.3 is used. Otherwise, the traditional translation method is utilized. Obviously, any CKY-based translation method can be used to generate translation candidates, such as BTG translation model and hierarchical phrase-based translation model. In this process, PAS span and non-PAS span are used equally for translating bigger spans. This is because bad PASs might harm the translation accuracy and the competition of PAS spans and non-PAS spans will help to choose good PASs.

For a specific span, we distinguish its translation candidates from different PASs by the two rule probabilities in sub-section 2.4 and the matching score in sub-section 3.1. These probabilities and scores are served as the PAS features for decoding. Their weights are tuned together with other features, such as language model. We call this translation system as *PAS transformation system*.

In the decoder, we can see that the translation candidates of PAS span are generated only by PAS transformation rules, while the traditional translation method also has its own way to translate the same PAS span. We believe that they complement each other because they perform translation from different angles. Thus, to capture this complementation, for the PAS span in the decoding hypergraph, we can use both our PAS-based translation method and the traditional translation method. This leads to a combination system which we call *PAS combination system*.

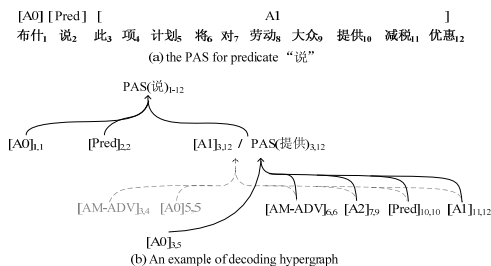


FIGURE 8 – An illustration of the decoding hypergraph. In the FIGURE, the PASs for predicate “提供 (provide)” are the result M1 and M2 in FIGURE 4(b). We omit the non-PAS spans here.

## 5 Experiment

### 5.1 Experimental Setup

The experiment is conducted on Chinese-to-English translation. The training data includes 260K bilingual sentence pairs<sup>8</sup>. To guarantee the accuracy of bilingual SRL, the length of each sentence is among 10 and 30 words. We use this data for both bilingual SRL and training the translation system. We first run GIZA++ and employ the *intersection* and *grow-diag-final-and (gdfa)* strategy respectively to produce symmetric word alignments. Then we use the intersection alignment to find the aligned predicates and adopt Zhuang and Zong (2010b)’s method to do bilingual SRL. After that, we refine the result in terms of the gdfa alignment and extract PAS transformation rules as described in section 2.

For machine translation, we train a 5-gram language model with the Xinhua portion of English Gigaword corpus and target part of training data. The development set and test set are the NIST evaluation test data (from 2003 to 2005). To get accurate SRL results, we also only extract sentences whose lengths are among 10 and 30 words. As a result, 595 sentences from NIST MT03 serve as the development set. 1,786 sentences from NIST MT04 and MT05 compose the test set. We perform SRL for the two sets by Zhuang and Zong (2010b)’s method. The translation quality is evaluated by case-insensitive BLEU-4 with shortest length penalty. The statistical significance test is performed by the re-sampling approach (Koehn, 2004). We employ our in-house BTG system used in (Zhang and Zong, 2009) to serve as our baseline translation method. We use PAS(BTG) to denote the PAS transformation system and PAS+BTG to represent the PAS combination system.

### 5.2 PAS Transformation Rules

In the training data, we acquire 226,968 aligned predicate pairs. From these predicate pairs, we extract 62,597 different simplified PAS transformation rules and then we extend them to 92,278 ones. Among the rules, 99.55% of their TPs can be binarized. Therefore, our decoding algorithm in sub-section 3.3 can be used in almost all cases. To detail our PAS transformation rules, we give the top 5 monotone rules and reordering rules respectively in TABLE 1.

<sup>8</sup> It is extracted from the LDC corpus. The LDC category number: LDC2000T50, LDC2002E18, LDC2003E07, LDC2004T07, LDC2005T06, LDC2002L27, LDC2005T10 and LDC2005T34.

Top 5 monotone rules			Top 5 reordering rules		
Pred	SP	TP	Pred	SP	TP
说(say)	[A0] <sub>1</sub> [Pred] <sub>2</sub> [A1] <sub>3</sub>	X <sub>1</sub> X <sub>2</sub> X <sub>3</sub>	提供(provide)	[A0] <sub>1</sub> [A2] <sub>2</sub> [Pred] <sub>3</sub> [A1] <sub>4</sub>	X <sub>1</sub> X <sub>3</sub> X <sub>4</sub> X <sub>2</sub>
认为(think)	[A0] <sub>1</sub> [Pred] <sub>2</sub> [A1] <sub>3</sub>	X <sub>1</sub> X <sub>2</sub> X <sub>3</sub>	支持(support)	[A0] <sub>1</sub> [AM-ADV] <sub>2</sub> [Pred] <sub>3</sub> [A1] <sub>4</sub>	X <sub>2</sub> X <sub>1</sub> X <sub>3</sub> X <sub>4</sub>
希望(hope)	[A0] <sub>1</sub> [Pred] <sub>2</sub> [A1] <sub>3</sub>	X <sub>1</sub> X <sub>2</sub> X <sub>3</sub>	说(say)	[A0] <sub>1</sub> [Pred] <sub>2</sub> [A1] <sub>3</sub>	X <sub>3</sub> X <sub>1</sub> X <sub>2</sub>
想(think)	[A0] <sub>1</sub> [Pred] <sub>2</sub> [A1] <sub>3</sub>	X <sub>1</sub> X <sub>2</sub> X <sub>3</sub>	表示(express)	[A0] <sub>1</sub> [A3] <sub>2</sub> [Pred] <sub>3</sub> [A1] <sub>4</sub>	X <sub>1</sub> X <sub>3</sub> X <sub>4</sub> X <sub>2</sub>
有(have)	[A0] <sub>1</sub> [Pred] <sub>2</sub> [A1] <sub>3</sub>	X <sub>1</sub> X <sub>2</sub> X <sub>3</sub>	举行(hold)	[A1] <sub>1</sub> [AM-LOC] <sub>2</sub> [Pred] <sub>3</sub>	X <sub>1</sub> X <sub>3</sub> X <sub>2</sub>

TABLE 1 – Top 5 monotone rules and reordering rules. The counts of monotone rules range from 5,101 to 1,745, and the counts of reordering rules range from 339 to 157.

Let us investigate the reordering rules first. The transformation rule for Chinese verb “提供(provide)” moves its argument [A2] behind [Pred] and [A1]. In general, [A2] is usually a prepositional phrase, which begins with a prepositional word, such as “为(for)” or “向(to)”. This is reasonable because we always move the prepositional phrase behind verb phrase during Chinese-to-English translation, just as FIGURE 2(a) shows. From the transformation rules, we can see that we reorder the arguments based on the entire PAS. This demonstrates that our PAS-based translation method is good at global reordering.

For the monotone rules, we can see that all top 5 rules focus on [A0], [Pred] and [A1]. This fact demonstrates that Chinese and English are mostly Subject-Verb-Object (SVO) languages. Therefore, during Chinese-to-English translation, we can maintain the main skeleton structure of sentences according to the monotone rules.

### 5.3 Translation Result

TABLE 2 illustrates the final translation results of our experiments. As we can see, our in-house BTG system outperforms Moses (Koehn et al., 2007) by 0.33 BLEU points, indicating that our BTG system is a strong baseline system. Moreover, from TABLE 2, we can see that system PAS(BTG) only improves the baseline BTG system slightly, by 0.38 BLEU points. However, the PAS+BTG system significantly outperforms the baseline BTG system by 1.14 BLEU points. This comparison means that the PAS can better play its role by combining with BTG model. We will conduct a deep analysis on these results in the next sub-section.

System	Test Set	n-gram precisions			
		1	2	3	4
Moses	32.42	74.91	41.86	24.4	14.43
BTG	32.75	74.39	41.91	24.75	14.91
PAS(BTG)	33.13	75.13	42.55	25.10	15.02
PAS+BTG	<b>33.89*</b>	74.98	43.17	25.91	15.72

TABLE 2 – Result of BTG system and our PAS-based translation method. The “\*” denotes that the result is significantly better than BTG ( $p < 0.01$ ).

### 5.4 Analysis and Discussion

According to our statistics, in the total 1,786 test sentences, there are 1,747 ones have involved in matching PAS transformation rules. However, only 386 sentences in system PAS(BTG) but 1,017 sentences in system PAS+BTG have utilized PASs to generate final translations. Why they have such great difference in the two systems? After analysis, there are two main reasons.

On one hand, decoding space is narrowed and limited by the rigid spans under the PAS-based framework. As we described in sub-section 3.3, we use a CKY-style algorithm to enlarge the decoding space. However, even so, a lot of spans are still ignored during decoding. Moreover, the predetermined spans of arguments also restrict the usage of phrase translation rules.

On the other hand, the accuracy of SRL is not high. To our best knowledge, the F-score of current monolingual Chinese SRL system is only about 80% on the Treebank data. Moreover, this evaluation focuses on arguments, rather than the entire PASs. We can imagine that it would reduce greatly on the non-well-formed training and test data. In addition, according to our statistics, there are 26,809 different matching PASs in the test set in total, in which 16,489 ones (61.5% of all) have a father PAS or child PAS. This means such PAS is an argument of a bigger PAS or carries an argument which is actually a smaller PAS, just as FIGURE 8 shows. This hierarchical structure magnifies the negative impact of bad PASs in system PAS(BTG). Many accurate PASs are thus ignored because of its bad father PAS or child PAS.

Due to the narrow decoding space and bad PASs, the comprehensive translation score of PASs' translation candidates would be too low to be utilized in system PAS(BTG). Therefore, numerous PASs are bypassed by the decoder and only a slight improvement is achieved by system PAS(BTG). To address this problem, we propose system PAS+BTG. It not only combines the decoding space of our PAS-based translation framework and BTG translation model, but also breaks up the close connection between father PAS and child PAS by introducing BTG model's translation candidates for PASs. At last, it achieves significant improvement over BTG system and more PASs in 1,017 sentences are utilized in the system.

<i>PAS(BTG)</i>	# PAS-Span-Covered-Rate (named as cover-rate)			
	[0,50%)	[50%,100%)	100%	total
	181	65	225	471
<i>PAS+BTG</i>	# PAS-Span-Covered-Rate (named as cover-rate)			
	[0,50%)	[50%,100%)	100%	total
	613	775	125	1613

TABLE 3 – Statistics about PAS spans used for generating the final best translations. In the TABLE, for example, column 2 of system PAS(BTG) denotes that 65 PASs covering 50%~100% words of source sentences are utilized in system PAS(BTG).

To verify our above analysis, we further give TABLE 3. As we can see, comparing with PAS(BTG), much more PASs are used in PAS+BTG (471 vs 1613). Moreover, the number of PASs in PAS(BTG) reduces when the *cover-rate* increases<sup>9</sup>, while the number for PAS+BTG grows. Just as we discussed above, this is because the big PAS in PAS(BTG) usually depends not only on itself, but also on its child PAS. Once the big PAS carries a bad child PAS, its translation would be also bad due to this child PAS. Therefore, the number of big PASs used in PAS(BTG) reduces. In contrast, the child PAS in PAS+BTG is only a choice but not essential for translating its father PAS. Hence, the number of big PASs used in PAS+BTG increases.

From TABLE 3, we can also see that most of the PASs cover more than 50% words of source sentences. We call these PASs as *sen-wide PAS*. In system PAS+BTG, the number of *sen-wide*

<sup>9</sup> There is an exception when the *cover-rate* is 100% in system PAS(BTG). This is because the 225 test sentences are fully covered by PASs. In system PAS(BTG), the translation of these sentences must be generated by the PAS spans whose *cover-rate* are 100%. Obviously, this is a rigid constraint. We relax this constraint in PAS+BTG system to ignore the bad PASs and 125 ones are kept for the final translation.

PASs is 900 (i.e., 775+125 in TABLE 3) and the number for system PAS(BTG) is 290 (i.e., 225+65 in TABLE 3). Each of these PASs belongs to one individual sentence because they all cover more than 50% words of the sentences. Consequently, 88.5% (900/1,017) sentences in PAS+BTG system and 75% (290/386) sentences in PAS(BTG) system have utilized these *sen-wide PASs*, by which the skeleton structure of sentences are well modeled for translation. Hence, we can conclude that our PAS-based translation method performs global reordering based on these *sen-wide PASs* and achieves improvements over the baseline BTG system.

BTG	联合国 [已经 对 印尼 政府] [加诸于 外国 部队 的] [期限 表示 关切] un [加诸于 foreign troops] [already in the indonesian government] [expressed concern over the period]
PAS(BTG)	联合国 [A0] [AM-ADV] [已经 对 印尼 政府 加诸于 外国 部队 的 期限 表示 关切] [A3] [Pred] [AI] [the united nations] has expressed concerns [over 加诸于 period for foreign troops to the indonesian government]
ref	联合国 已经 对 印尼 政府 加诸于 外国 部队 的 期限 表示 关切 [the united nations] has expressed concern [over the deadline the indonesian government imposed on foreign troops]
BTG	海南 2005年 将 继续 增加 [对 公共 服务 和 社会 事业 基础 设施 投资] hainan 2005 will [continue to increase] [investment in public services, infrastructure and social undertakings]
PAS(BTG)	海南 2005年 将 继续 增加 对 公共 服务 和 社会 事业 基础 设施 投资 [A0] [AM-TMP] [AM-ADV] [Pred] [AI] hainan will [continue to increase] [investment in public services, infrastructure and social undertakings] [in 2005]
ref	海南 2005年 将 继续 增加 对 公共 服务 和 社会 事业 基础 设施 投资 [hainan province] will [continue to increase] [its investment in the public services and social services infrastructures] [in 2005]

TABLE 4 – Two translation examples of BTG system, PAS(BTG) system, and reference.

We further give two translation examples in TABLE 4 to specially show the effectiveness of our PAS-based translation method. For the first example, BTG system chooses a wrong manner to segment the big prepositional phrase “对 印尼 政府 加诸于 外国 部队 的 期限” into 3 parts. This is because BTG system only tries to get a translation with an average distribution of phrase segmentation. Moreover, since its translation model does not consider any information of sentence structure, it wrongly segments the test sentence and produces a bad translation. Conversely, our PAS(BTG) system segments the sentence based on its PAS. Since a correct PAS denotes the skeleton structure of the sentence, it performs both reasonable sentence segmentation and better global phrase reordering for translation. Furthermore, in the second example, our PAS-based method successfully recognizes the [AM-TMP] argument “2005年” and move it to the end of sentence. However, the BTG system only performs translation without any reordering.

## 6 Related Work

Previous work utilizing PAS in SMT can be roughly categorized into three directions.

One direction is to do pre-processing or post-processing. Komachi and Matsumoto (2006) and Wu et al. (2011) used PAS-based heuristic rules and automatic rules respectively to pre-order the

input sentences. Wu and Fung (2009b) performed SRL on the outputs of phrase-based system Moses and then reordered the achieved semantic roles to match the roles of input sentences.

Some other works tried to design proper PAS-based features and integrate them into decoder. Liu and Gildea (2010) projected source-side PASs to target side via word alignment and designed a “Semantic Role Re-ordering” feature and a “Deleted Roles” feature for tree-to-string model. Xiong et al. (2012) adopted semantic features to translate verbal predicates and predict the relative position between predicates and arguments.

Some other works focused on utilizing semantic roles to refine the non-terminals of syntax-based translation model. Liu and Gildea (2008) substituted the syntactic labels with semantic roles or combined them together for a tree-to-string model. Aziz et al., (2011) used semantic roles and base-phrase tags to create shallow semantic trees. Gao and Vogel (2011) used target side semantic roles to create SRL-aware non-terminals for hierarchical phrase-based model.

Our work is different from the existing work in the following aspects: (1) we induce PAS transformation rules to model the interrelation between source-side PAS and its target counterpart; (2) we utilize multiple SRL results to alleviate the negative impact of bad PASs; (3) we design a CKY algorithm to translate the entire PAS according to the target-side-like PAS. The algorithm can be easily integrated with any CKY-based decoder to generate better translation hypotheses.

## **Conclusion and Perspectives**

In this paper, we focus on building a PAS-based translation framework for modeling semantic structures in translation model. We first extract PAS transformation rules to model the intrinsic connection between source-side and target-side PASs. Then we perform machine translation in 3 steps: *PAS acquisition*, *transformation* and *translation*. Experimental results demonstrate that our PAS-based translation method improves the translation performance significantly.

Our method improves the translation performance in the following aspects: (1) take advantage of PAS, which keeps consistency well across languages; (2) use PAS transformation rules to perform global reordering in a skeleton scenario; (3) design reasonable strategies to exert the merit of PAS to segment sentences for translation; (4) the PAS-based translation framework can be easily integrated with any CKY-based translation models to generate better translations. In all, the translation process of our PAS-based translation method is similar to human translation to a great extent and it still has much room to improve with the upgrading of SRL performance. We believe it would be a big step towards semantics-based translation model.

In the next step, we will conduct further experiments on other language pairs to demonstrate the effectiveness of our PAS translation method, especially the translation between an SVO language and an SOV language. In addition, we also will utilize the target-side semantic roles to evaluate the quality of translation candidates and the structural integrity of translations.

## **Acknowledgments**

The research work has been funded by the Natural Science Foundation of China under Grant No. 6097 5053 and the Hi-Tech Research and Development Program (“863” Program) of China under Grant No. 2012AA011102 and 2011AA01A207. We would also like to thank Tao Zhuang for the help in generating bilingual SRL results, and the anonymous reviewers for their valuable comments.

## References

- Aziz, W., Rios, M., and Specia, L. (2011). Shallow semantic trees for smt. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT '11*, pages 316–322, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Bikel, D. (2004). Intricacies of Collins parsing model. *Computational Linguistics*, 30(4):480–511.
- Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.
- Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
- Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 205–208, Sapporo, Japan. Association for Computational Linguistics.
- Fung, P., Wu, Z., Yang, Y. and Wu, D. (2006). Automatic learning of chinese english semantic structure mapping. In *IEEE/ACL 2006 Workshop on Spoken Language Technology (SLT 2006)*, Aruba, December.
- Fung, P., Wu, Z., Yang, Y. and Wu, D. (2007). Learning bilingual semantic frames: shallow semantic sarsing vs. semantic sole projection. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation*, pages 75–84.
- Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia. Association for Computational Linguistics.
- Gao, Q. and Vogel, S. (2011). Utilizing target-side semantic role labels to assist hierarchical phrase-based machine translation. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, SSST-5*, pages 107–115, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.
- Koehn, P., Och, F. J. and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 58–54, Edmonton, Canada, May-June.
- Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In Lin, D. and Wu, D., editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007).

- Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Komachi, M. and Matsumoto, Y. (2006). Phrase reordering for statistical machine translation based on predicate-argument structure. In *Proceedings of the International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation*, pages 77–82.
- Liu, D. and Gildea, D. (2008). Improved tree-to-string transducer for machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 62–69, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Liu, D. and Gildea, D. (2010). Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 716–724, Beijing, China. Coling 2010 Organizing Committee.
- Liu, Y., Liu, Q., and Lin, S. (2006). Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia. Association for Computational Linguistics.
- Marcu, D., Wang, W., Echihiabi, A., and Knight, K. (2006). Spmt: Statistical machine translation with syntactified target language phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 44–52, Sydney, Australia. Association for Computational Linguistics.
- Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.
- Och, F. J. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417–449.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia. Association for Computational Linguistics.
- Stolcke, A. (2002). Srilm – an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado, USA, September.
- Wang, W., Knight, K., and Marcu, D. (2007). Binarizing syntax trees to improve syntax-based machine translation accuracy. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 746–754, Prague, Czech Republic. Association for Computational



Linguistics.

Wu, D. and Fung, P. (2009a). Can semantic role labeling improve smt. In *Proceedings of the 13th Annual Conference of the EAMT*, pages 218–225, Barcelona, May.

Wu, D. and Fung, P. (2009b). Semantic roles for smt: A hybrid two-pass model. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 13–16, Boulder, Colorado. Association for Computational Linguistics.

Wu, S. and Palmer, M. (2011). Semantic mapping using automatic word alignment and semantic role labeling. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, SSST-5*, pages 21–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wu, X., Sudoh, K., Duh, K., Tsukada, H., and Nagata, M. (2011). Extracting pre-ordering rules from predicate-argument structures. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 29–37, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Xiong, D., Liu, Q., and Lin, S. (2006). Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, Sydney, Australia. Association for Computational Linguistics.

Xiong, D., Zhang, M., and Li, H. (2011). A maximum-entropy segmentation model for statistical machine translation. *IEEE Transactions on Audio, Speech and Language Processing*, 19(8):2494–2505.

Xiong, D., Zhang, M., and Li, H. (2012). Modeling the translation of predicate-argument structure for smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 902–911, Jeju Island, Korea. Association for Computational Linguistics.

Xue, N. (2008). Labeling chinese predicates with semantic roles. *Computational Linguistics*, 34(2): 225-255.

Zhai, F., Zhang, J., Zhou, Y., and Zong, C. (2011). Simple but Effective Approaches to Improving Tree-to-Tree Model. *MT-Summit-11*. pages 261-268.

Zhang, H., Zhang, M., Li, H., and Chng, E. S. (2010). Non-isomorphic forest pair translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Cambridge, MA. Association for Computational Linguistics.

Zhang, J., Zhai, F., and Zong, C. (2011). Augmenting string-to-tree translation models with fuzzy use of source-side syntax. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 204–215, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Zhang, J. and Zong, C. (2009). A framework for effectively integrating hard and soft syntactic rules into phrase based translation. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pages 579–588, Hong Kong. City University of Hong Kong.

Zhuang, T. and Zong, C. (2010a). Joint inference for bilingual semantic role labeling. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 304–314, Cambridge, MA. Association for Computational Linguistics.

Zhuang, T. and Zong, C. (2010b). A minimum error weighting combination strategy for chinese semantic role labeling. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1362–1370, Beijing, China. Coling 2010 Organizing Committee.