

# Well-Formed Dependency to String translation with BTG Grammar

Xiaoqing Li<sup>†</sup>, Kun Wang<sup>†</sup>, Dakun Zhang<sup>‡</sup>, Jie Hao<sup>‡</sup>

<sup>†</sup> National Laboratory of Pattern Recognition, Institute of Automation,  
Chinese Academy of Sciences

<sup>‡</sup> Toshiba (China) R&D Center

{xqli, kunwang}@nlpr.ia.ac.cn

{zhangdakun, haojie}@toshiba.com.cn

## Abstract

This paper proposes a well-formed dependency to string translation model with BTG grammar. By enabling the usage of well-formed sub-structures and allowing flexible reordering of them, our approach is effective to relieve the problems of parsing error and flatness in dependency structure. To utilize the well-formed dependency rules during decoding, we adapt the tree traversal decoding algorithm into a bottom-up CKY algorithm. And a lexicalized reordering model is used to encourage the proper combination of two neighbouring blocks. Experiment results demonstrate that our approach can effectively improve the performance by more than 2 BLEU score over the baseline.

## 1 Introduction

Due to the merits of holding shallow semantic information and cross-lingual consistency (Fox, 2002), dependency grammar has attracted much attention in the field of machine translation (Lin, 2004; Quirk et al., 2005; Ding and Palmer, 2005; Shen et al., 2008; Xie et al., 2011).

The dependency-to-string model (Xie et al., 2011) falls into the paradigm of "translation after understanding", which tries to understand the structure and meaning of source text. However, there are two typical problems for this approach. One is that the model is prone to be affected by parsing errors. Dependency-to-string model adopts a unique source side tree structure as fixed input and constructs the output by converting each sub-structure into target

side. If there is some errors in the source dependency tree, for example, a prepositional subtree is attached to a wrong head, the model can hardly recover the error. Figure 1(a) shows another parsing error, in the correct parsing result, "与北韩...国家" should forms a subtree with "国家" as the head, and then this subtree is dominant by "之一".

The other problem is that dependency structure is too flat for the translation task. Since dependency-to-string model requires a head and all its dependents to be translated as a whole, the flatness of the structure will make rules difficult to be matched during decoding. Furthermore, it will also lower the robustness of translation rules, since many giant and low-frequency rules will be extracted. Figure 1(b) shows an example of the flat structure. The head word "提供" has five dependents in the structure. If no rule can be matched, only the glue rule can be applied. As a result, the prepositional subtree "为了...义务" cannot be correctly reordered to the end in the target side.

Existing solutions for the above problems include (Meng et al., 2013; Xie et al., 2014). The former incorporates phrasal nodes of constituency trees in the source side of translation rules, and the latter modifies translation rules during decoding to allow the usage of phrases which are compatible with well-formed structures. Since these two approaches still adopt head-dependent structure as the backbone of the translation rule, the freedom of generating translation candidates is limited.

On the contrary, we propose to use BTG grammar to combine the translations of two adjacent well-formed structures. To incorporate the BTG rules

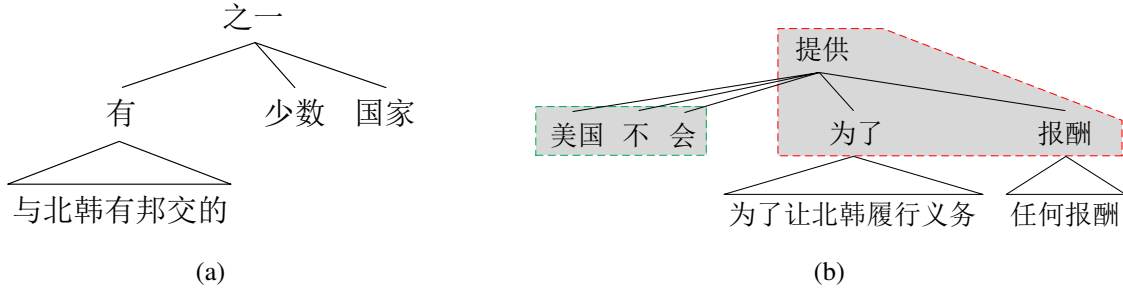


Figure 1: Examples of parsing error (a) and flatness (b)

into the model, we adapt the tree traversal decoding algorithm into a bottom-up CKY algorithm. Large scale experiments show that our approach can improve the performance by more than 2 BLEU score over the baseline, and it is also superior to the two approaches mentioned above.

## 2 Background

We briefly review the dependency to string model and the BTG grammar in this section, which are the bases of our proposed model.

### 2.1 Dependency-to-String Model

The dependency-to-string model proposed by (Xie et al., 2011) translates a source dependency tree by applying head-dependents translation rule at each head node in a recursive way. A head-dependents translation rule consists of a head-dependents fragment in the source side and its translation correspondence in the target side. The rule  $r1$  in Figure 2 is an example of their translation rule. This rule specifies the translation of the head node "提供" and leaf nodes "布什", and also the reordering relation of the non-terminal nodes, including the internal node "为" and the generalized internal node "优惠". The word or POS tag at each non-terminal node in the rule describes its matching condition. For example,  $X2:NN$  in  $r1$  means the second non-terminal must be a noun while matching this rule. In principle, all nodes, i.e., head, internal and leaf nodes in the dependency tree, can be generalized to their POS tags (or other categories) to relieve data sparsity.

By including a head and all its dependents into one rule, the dependency-to-string model is good at long distance reordering. However, this structure is not robust enough due to parsing errors and flatness.

### 2.2 BTG Grammar

Bracketing transduction grammar (BTG) (Wu, 1997) is a special case of synchronous context free grammar. There are only two types of rules in this grammar:

$$X \rightarrow [X^1, X^2] \mid X \rightarrow \langle X^1, X^2 \rangle$$

$$X \rightarrow x/y$$

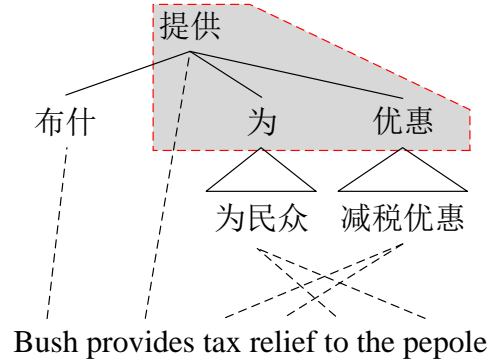
The first type of rule is used to merge the translations of two neighbouring blocks  $X^1$  and  $X^2$  with monotone or swap order, and the second type of rule is used to translate source phrase  $x$  in to target phrase  $y$ . Due to its simplicity and effectiveness of modeling bilingual correspondence, BTG grammar is widely used translation modeling (Xiong et al., 2006; Li et al., 2013), word alignment (Zhang and Gildea, 2005; Haghighi et al., 2009; Pauls et al., 2010), translation system combination (Karakos et al., 2008), etc.

### 3 Well-Formed Dependency to String Model

In this section, we describe our well-formed dependency to string model with BTG grammar, and explain how it relieves the problems of parsing error and flatness.

#### 3.1 Modified Well-formed structure

Similar to (Shen et al., 2008), we define two kinds of well-formed dependency structures, i.e., fixed structure and floating structure. Fixed structure consists of the heads of a sequence of sibling trees and the common head of these trees; and floating structure consists of the heads of a sequence of sibling trees without their common head. The difference between



$r1$  : 布什 X1:为 提供 X2:NN ||| Bush provides X2 X1  
 $r2$  : X1:为 提供 X2:NN ||| provides X2 X1  
 Head node in the rules are marked with bold face. NN is the POS tag of the source word "优惠".

Figure 2: examples of dependency to string rule ( $r1$ ) and well-formed dependency to string rule ( $r2$ )

our definition and that in Shen et al., 2008 is that we only include the heads of subtrees in our structure, while they include the whole subtrees. The shadowed part with red box in Figure 1(b) is an example of fixed structure, which consists of a head "提供" and the heads of two continuous sibling trees "为了" and "报酬". And the shadowed part with green box in Figure 1(b) is an example of floating structure, which consists of the heads of three continuous sibling trees "美国", "不" and "会".

Given a sentence  $w_1w_2...w_n$ , let  $d_i$  denote the head index of  $w_i$ , our fixed and floating structure can be formally defined as follows,

**Definition 1**

A fixed structure  $f_{h,C}$  with head  $h$  and children  $C$ , where  $h \in [1, n]$  and  $C \subseteq \{1, \dots, n\}$ , is a two-level tree fragment which satisfies the following conditions:

- $\forall k \in C, d_k = h$
- $\forall \min(C) \leq k \leq \max(C), d_k \neq h$

**Definition 2** A floating structure  $f_C$  with children  $C$ , where  $C \subseteq \{1, \dots, n\}$ , is a one level tree fragment which satisfies the following conditions:

- $\exists h, \forall k \in C, d_k = h$
- $\forall \min(C) \leq k \leq \max(C), d_k \neq h$

**3.2 Well-Formed Dependency-to-String Rule**

Our well-formed dependency to string translation rule consists of a well-formed dependency structure in the source side and its translation correspondence in the target side. This definition extends the rule proposed in (Xie et al., 2011) to cover all well-formed dependency structures in the source side, rather than using complete head-dependents structures only. The rule  $r2$  is an examples of our translation rule. Compared with  $r1$ , this rule does not contain "布什" in the source side (and its translation in the target side). Since it contains less context, this rule is more flexible to be applied during decoding. For example, if "布什" is replaced with a pronoun "他" in a testing sentence, this rule can still be applied. However,  $r1$  cannot be applied in this case even if it is generalized, since the POS tag of "他" does not match that of "布什".

Our translation rules can be extracted from aligned dependency tree and string pair by traversing the tree and enumerating the well-formed structure at each node. Following previous work(Koehn et al., 2003; Galley et al., 2004; Chiang, 2005), we impose alignment constraint for rule extraction. The intuition is that words in the one side (source/target) cannot be aligned to words outside the other side, and the word alignment within non-terminals also need to satisfy this constraint.

Formally, for a non-terminal node  $n$ , we define node span  $nsp(n)$  as the closure of the indexes of those words that  $n$  is aligned to, and sub-tree span  $ssp(n)$  as the closure of node spans of all the nodes in the subtree rooted with  $n$ . These two spans are set to  $\phi$  for terminals. In addition, we use  $N_s$  to denote the set of all the terminal indexes in the source side, and  $N_t$  to denote the set of all the terminal indexes in the target side. Function  $a(\cdot)$  is used to get the indexes of the aligned words for a give word. Then the alignment constraint can be described as follows,

- $\forall k \in N_s, a(k) \in N_t$
- $\forall k \in N_t, a(k) \in N_s$
- $nsp(head) \cap_{n \in children} ssp(n) = \Phi$

A minor difference in our constraint with (Xie et al., 2011) is that we allow the alignment of terminals to be overlaped. For example, in Figure 1(b), if the

two terminals ”不” and ”会” align to a single target word ”won’t”, we consider the alignment constraint is satisfied, while they consider it as invalid.

### 3.3 Apply Dependency to String Rules with BTG Rules

We use the examples in Figure 1 to illustrate how the well-formed dependency to string rules together with BTG rules can be used to overcome the problems of parsing error and flatness. A plausible derivation for the example in Figure 1(a) is shown in Figure 3, in which the subtree ”与...的”, the floating structure ”少数国家” and the head node is translated first, then the translations of the first two parts can be combined with a BTG rule of swap order. The final translation can be achieved by applying another BTG rule of swap order to the translation just obtained and the translation of ”之一”. Note that this is not the only derivation that can lead to a correct translation. We can also combine the translation of floating structure ”少数国家” and the head node first, then combine with the translation of the first subtree. Similarly, for the example in Figure 1(b), we can first translate the floating structure ”美国不会” and the fixed structure ”为了...提供...报酬”, then combine them with an BTG rule of monotone order to produce the final translation.

## 4 Decoding

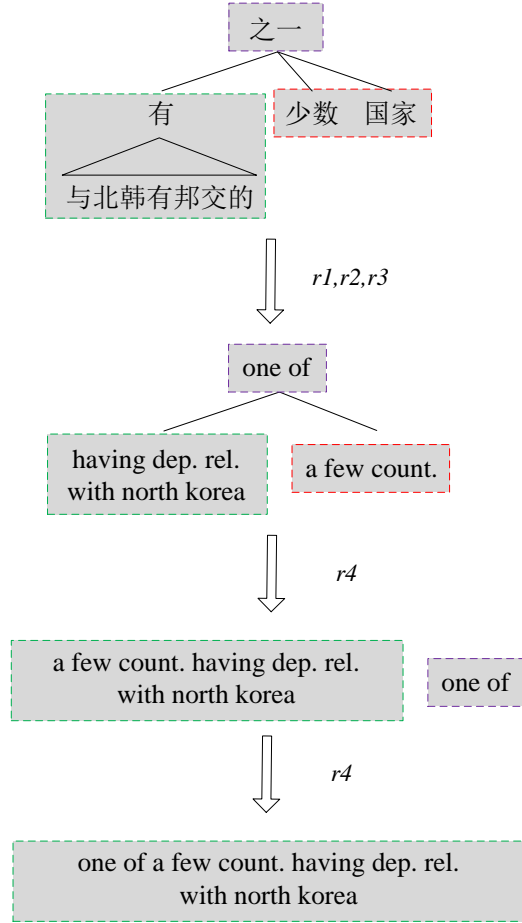
### 4.1 Model

We use the standard log-linear model (Och and Ney, 2002) to score the translation hypothesis during decoding. For a specific derivation  $d$  that converts a source dependency tree  $T$  into a target string  $s$ , the score of  $d$  will be,

$$P(d) \propto \prod_i \phi_i(d)^{\lambda_i}$$

where  $\phi_i$  are features defined on derivations and  $\lambda_i$  are corresponding weight for each feature. The features adopted in this paper include bidirectional translation probabilities, bidirectional lexical weights, language model, rule penalty, word penalty and reordering probability for BTG rules.

For the last feature, we use a maximum entropy model to estimation the probability and the same



- $r1$ : 之一 ||| one of
- $r2$ : 与北韩有邦交的 ||| having dep. rel. with north korea
- $r3$ : 少数国家 ||| a few countries
- $r4$ :  $X1 X2$  |||  $X2 X1$

Figure 3: a plausible derivation with well-formed dependency to string and BTG grammar for the example in Figure 1(a)

features in (Xiong et al., 2006) are adopted, including beginning and ending words in the two blocks to be reordered, from both the source and target side. So there are eight activated features in total for each instance.

### 4.2 Decoding Algorithm

The decoding algorithm is described in Algorithm 1. The algorithm begins by translating each word in the sentence, then proceed to translate larger spans

**Algorithm 1:** CKY decoding algorithm with well formed dependency to string rules

---

**Input:** Source dependency tree  $T$  with  $N$  words  
**Output:** Target translation

```

for  $span: 1 \rightarrow N$  do
  for  $start: 1 \rightarrow N+1-span$  do
    generate initial candidates with well formed dependency to string rule;
    for  $span\_lhs: 1 \rightarrow span-1$  do
      if  $\{span\_lhs, span\_rhs, span\} \subseteq well\text{-formed structure}$  then
        generate initial candidates with BTG rules;
      end
    end
    generate KBEST candidates with cube pruning;
  end
end
return the top candidate over the whole sentence as output;

```

---

in an bottom up manner. When translating a span compatible with a well-formed structure, there are two ways to generate translation candidates. One is based on fixed or floating rules which covers the whole span, and the other is combining the translation candidates in two sub-spans with BTG rule of monotone or swap order. The two sub-spans also need to be compatible with well-formed structure. The cube pruning algorithm (Chiang, 2007; Huang and Chiang, 2007) is used to expand the initial candidates until Kbest candidates have been generated. Finally, the top candidate over the whole sentence will be returned as output.

## 5 Experiments

We evaluate the performance of our model on Chinese to English translation. And we re-implement the dependency to string model for performance comparison.

### 5.1 Data preparation

Two sets of training data are adopted in our experiments. The smaller one consists of 270k sentence pairs, and the larger one consists of 2.1M sentence pairs. All the training data comes from the LDC corpus<sup>1</sup>. And we use NIST 02 test set as our development set, NIST 03 and 04 test set as our test

<sup>1</sup>Including LDC2000T50, LDC2002E18, LDC2003E07, LDC2003E14, LDC200407, LDC2005T06, LDC2002L27, LDC2005T10 and LDC 2005T34.

set. The case insensitive NIST BLEU-4 metric (Papineni et al., 2002) is adopted for evaluation. We use the SRILM toolkit to train a 5-gram language model with Kneser-Ney smoothing on the Xinhua portion of the Gigaword corpus.

The source side of the training and dev/test set are segmented with our in house segmentation tool (Wang et al., 2010). And they are parsed with Stanford Parser (De Marneffe et al., 2006), which also generates POS tag for each word. The dependency relations on edges are not used in this work.

Word alignments are obtained with our in house tool (Wang and Zong, 2013), which takes dependency cohesion constraints into consideration while doing word alignments. And we use the MaxEnt toolkit<sup>2</sup> to estimate the context sensitive reordering probability for BTG rules. The weights of the features are tuned with MERT (Och, 2003) to maximize the BLEU score on the development set.

### 5.2 Results

The strength of our model lies in two aspects. First, our translation unit is more fine-grained than that in the original dependency to tree model, which enables the translation of many linguistically plausible phrases; second, we allow flexible reorderings for adjacent blocks under the guide of context information. To check whether these two points hold, two sets of experiments are conducted in line. Initially,

<sup>2</sup><https://github.com/lzhang10/maxent>

System	02 (dev)	03	04	Average
dep2str	33.50	31.92	32.59	32.67
wf-d2s (mono)	35.03	33.31	34.50	34.28
wf-d2s	<b>35.86</b>	<b>34.04</b>	<b>35.20</b>	<b>35.03</b>

Table 1: Effects of applying well-formed dependency to string rules and allowing flexible reordering. The system wf-d2s (mono) denotes our well-formed dependency to string model with monotone reordering, and wf-d2s denotes our model with flexible reordering of two directions.

System	02 (dev)	03	04	Average
dep2str	35.24	34.45	34.50	34.73
wf-d2s	<b>37.07*</b>	<b>36.38*</b>	<b>37.01*</b>	<b>36.82</b>

Table 2: Experiment results with small and large training data. The "\*" denotes that the results are significantly better than the baseline (dep2str) system ( $p < 0.01$ ).

we only allow BTG rule with monotone order, i.e. translation of each well-formed structure are concatenated sequentially, which is equivalent to glue rule. Then BTG rules with both orders are enabled, with context sensitive reordering module. We conduct the experiments with the small training data set. The results are shown in Table 1. Compared with dependency to string rules, applying well-formed dependency to string rules significantly improves the performance by more than 1.5 BLEU score on average. If flexible reordering is further allowed, additional improvement of 0.7 BLEU score can be achieved.

Table 2 shows the performance of our model with large training set. Experiment results show that our model keeps its edge even with large training data. On average, more than 2 point in BLEU score are gained over the baseline. This improvement is much larger than (Meng et al., 2013) and (Xie et al., 2014). Both of them report improvement of about 0.9 point in BLEU score over the baseline on their dataset.

## 6 Related Work

The work that is most similar to ours is (Xie et al., 2014). However, there are several significant differences between these two work. First They incorporate well-formed dependency rules during decoding by modify the matched dependency rules "on the fly". For example, assume there is a matched rule "X1:NR X2:AD X3:VV X1:为了提供 X2:报酬||| X1 X2 X3 provide X5 X4" for the head-dependents structure in Figure 1 (b). in order to use the phrase "美国不会||| us won't" during decoding, they will compress the three nodes into one pseudo node "NR\_AD\_VV". Then the above rule will become "X1:NR\_AD\_VV X2:为了\* 提供 X3:报酬||| X1 provide X3 X2". This new rule will inherit the translation probabilities from the original rule. In the case that there is no matched rule or the probability estimation is unreliable due to sparsity, this method won't work well. Another difference is that they only use phrasal rules corresponding to well formed dependency structures, while we allow variables to be contained in the well-formed dependency rules.

The two problems of parsing error and flatness also exist in constituency tree . In order to make full use of the sub-structures, there have been a lot of work, including tree sequence to string translation (Liu et al., 2007), tree binarization (Zhang et al., 2006), forest-based translation (Mi et al., 2008) and fuzzy rule matching (Zhang et al., 2011).

## 7 Conclusion and Future Work

In this work, we propose a well-formed dependency to string model to address the problems of parsing error and flatness. By introducing translation rules corresponding to well-formed sub-structures, we are able to learn more reliable translation equivalents. During decoding, we propose to use BTG grammar with lexicalized reordering to combine translations of two neighbouring well-formed structures, which is more flexible than previous work. Experiment results demonstrate that our model can significantly improve translation performance.

Although our model is more flexible to generate translation candidates, it also brings more challenges to model translation quality. In the future, we will explore more powerful features to better score the translation candidates.

## Acknowledgments

This research work was funded by the Natural Science Foundation of China under Grant No. 61402478. The authors would like to thank Keh-Yih Su for insightful discussions.

## References

- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- David Chiang. 2007. Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.
- Marie-Catherine De Marneffe, Bill MacCartney, and Christopher Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.
- Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 541–548, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Heidi Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 304–311. Association for Computational Linguistics, July.
- Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.
- Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised itg models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 923–931, Suntec, Singapore, August. Association for Computational Linguistics.
- Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic, June. Association for Computational Linguistics.
- Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine translation system combination using itg-based alignments. In *Proceedings of ACL-08: HLT, Short Papers*, pages 81–84, Columbus, Ohio, June. Association for Computational Linguistics.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.
- Peng Li, Yang Liu, and Maosong Sun. 2013. Recursive autoencoders for ITG-based translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 567–577, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Dekang Lin. 2004. A path-based transfer model for machine translation. In *Proceedings of Coling 2004*, pages 625–630, Geneva, Switzerland, Aug 23–Aug 27. COLING.
- Yang Liu, Yun Huang, Qun Liu, and Shouxun Lin. 2007. Forest-to-string statistical translation rules. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 704–711, Prague, Czech Republic, June. Association for Computational Linguistics.
- Fandong Meng, Jun Xie, Linfeng Song, Yajuan Lü, and Qun Liu. 2013. Translation with source constituency and dependency trees. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1066–1076, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of ACL-08: HLT*, pages 192–199, Columbus, Ohio, June. Association for Computational Linguistics.
- Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.
- Franz Josef Och. 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, July. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, July. Association for Computational Linguistics.
- Adam Pauls, Dan Klein, David Chiang, and Kevin Knight. 2010. Unsupervised syntactic alignment with inversion transduction grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 118–126, Los Angeles, California, June. Association for Computational Linguistics.
- Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June. Association for Computational Linguistics.
- Zhiguo Wang and Chengqing Zong. 2013. Large-scale word alignment using soft dependency cohesion constraints. *Transactions of the Association for Computational Linguistics*, 1:291–300.
- Kun Wang, Chengqing Zong, and Keh-Yih Su. 2010. A character-based joint model for chinese word segmentation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1173–1181, Beijing, China, August. Coling 2010 Organizing Committee.
- Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403.
- Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 216–226, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Jun Xie, Jinan Xu, and Qun Liu. 2014. Augment dependency-to-string translation with fixed and floating structures. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2217–2226, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Deyi Xiong, Qun Liu, and Shouxun Lin. 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, July. Association for Computational Linguistics.
- Hao Zhang and Daniel Gildea. 2005. Stochastic lexicalized inversion transduction grammar for alignment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 475–482, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 256–263, New York City, USA, June. Association for Computational Linguistics.
- Jiajun Zhang, Feifei Zhai, and Chengqing Zong. 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., July. Association for Computational Linguistics.