On the Elements of an Accurate Tree-to-String Machine Translation System

Graham Neubig, Kevin Duh
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma-shi, Nara, Japan
{neubig,kevinduh}@is.naist.jp

Abstract

While tree-to-string (T2S) translation theoretically holds promise for efficient, accurate translation, in previous reports T2S systems have often proven inferior to other machine translation (MT) methods such as phrase-based or hierarchical phrase-based MT. In this paper, we attempt to clarify the reason for this performance gap by investigating a number of peripheral elements that affect the accuracy of T2S systems, including parsing, alignment, and search. Based on detailed experiments on the English-Japanese and Japanese-English pairs, we show how a basic T2S system that performs on par with phrase-based systems can be improved by 2.6-4.6 BLEU, greatly exceeding existing state-of-the-art methods. These results indicate that T2S systems indeed hold much promise, but the above-mentioned elements must be taken seriously in construction of these systems.

1 Introduction

In recent years, syntactic parsing is being viewed as an ever-more important element of statistical machine translation (SMT) systems, particularly for translation between languages with large differences in word order. There are many ways of incorporating syntax into MT systems, including the use of string-to-tree translation (S2T) to ensure the syntactic well-formedness of the output (Galley et al., 2006; Shen et al., 2008), tree-to-string (T2S) using source-side parsing as a hint during the translation process (Liu et al., 2006), or pre- or post-ordering to help compensate for reordering problems experienced by non-syntactic methods such as phrase-based MT (PBMT) (Collins et al., 2005; Sudoh et al., 2011). Among these, T2S translation has a number of attractive theoretical properties, such as joint consideration of global reordering and lexical choice while maintaining relatively fast decoding times.

However, building an accurate T2S system is not trivial. On one hand, there have been multiple reports (mainly from groups with a long history of building T2S systems) stating that systems using source-side syntax greatly out-perform phrase-based systems (Mi et al., 2008; Liu et al., 2011; Zhang et al., 2011; Tamura et al., 2013). On the other hand, there have been also been multiple reports noting the exact opposite result that source-side syntax systems perform worse than Hiero, S2T, PBMT, or PBMT with pre-ordering (Ambati and Lavie, 2008; Xie et al., 2011; Kaljahi et al., 2012). In this paper, we argue that this is due to the fact that T2S systems have the potential to achieve high accuracy, but are also less robust, with a number of peripheral elements having a large effect on translation accuracy.

Our motivation in writing this paper is to provide a first step in examining and codifying the more important elements that make it possible to construct a highly accurate T2S MT system. To do so, we perform an empirical study of the effect of parsing accuracy, packed forest input, alignment accuracy, and search. The reason why we choose these elements is that past work that has reported low accuracy for T2S systems has often neglected to consider one or all of these elements.

As a result of our tests on English-Japanese (en-ja) and Japanese-English (ja-en) machine translation, we find that a T2S system not considering these elements performs only slightly better than a standard PBMT system. However, after accounting for all these elements we see large increases of accuracy, with the final system greatly exceeding not only standard PBMT, but also state-of-the-art methods based on syntactic pre- or post-ordering.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 143–149,
Baltimore, Maryland, USA, June 23-25 2014.

2 Experimental Setup

2.1 Systems Compared

In our experiments, we use a translation model based on T2S tree transducers (Graehl and Knight, 2004), constructed using the Travatar toolkit (Neubig, 2013). Rules are extracted using the GHKM algorithm (Galley et al., 2006), and rules with up to 5 composed minimal rules, up to 2 non-terminals, and up to 10 terminals are used.

We also prepare 3 baselines not based on T2S to provide a comparison with other systems in the literature. The first two baselines are standard systems using PBMT or Hiero trained using Moses (Koehn et al., 2007). We use default settings, except for setting the reordering limit or maximum chart span to the best-performing value of 24. As our last baselines, we use two methods based on syntactic pre- or post-ordering, which are state-of-the-art methods for the language pairs. Specifically, for en-ja translation we use the head finalization pre-ordering method of (Isozaki et al., 2010b), and for ja-en translation, we use the syntactic post-ordering method of (Goto et al., 2012). For all systems, T2S or otherwise, the language model is a Kneser-Ney 5-gram, and tuning is performed to maximize BLEU score using minimum error rate training (Och, 2003).

2.2 Data and Evaluation

We perform all of our experiments on en-ja and ja-en translation over data from the NTCIR PatentMT task (Goto et al., 2011), the most standard benchmark task for these language pairs. We use the training data from NTCIR 7/8, a total of approximately 3.0M sentences, and perform tuning on the NTCIR 7 dry run, testing on the NTCIR 7 formal run data. As evaluation measures, we use the standard BLEU (Papineni et al., 2002) as well as RIBES (Isozaki et al., 2010a), a reordering-based metric that has been shown to have high correlation with human evaluations on the NTCIR data. We measure significance of results using bootstrap resampling at $p < 0.05$ (Koehn, 2004). In tables, bold numbers indicate the best system and all systems that were not significantly different from the best system.

2.3 Motivational Experiment

Before going into a detailed analysis, we first present results that stress the importance of the elements described in the introduction. To do so, we compare the 3 non-T2S baselines with two T2S systems that vary the settings of the parser, alignment, and search, as described in the following Sections 3, 4, and 5. The first system "T2S-all" is a system that uses the worst settings[1] for each of these elements, while the second system "T2S+all" uses the best settings.[2] The results for the systems are shown in Table 1.

	en-ja		ja-en	
System	BLEU	RIBES	BLEU	RIBES
PBMT	35.84	72.89	30.49	69.80
Hiero	34.45	72.94	29.41	69.51
Pre/Post	36.69	77.05	29.42	73.85
T2S-all	36.23	76.60	31.15	72.87
T2S+all	**40.84**	**80.15**	**33.70**	**75.94**

Table 1: Overall results for five systems.

The most striking result is that T2S+all significantly exceeds all of the baselines, even including the pre/post-ordering baselines, which provide state-of-the-art results on this task. The gains are particularly striking on en-ja, with a gain of over 4 BLEU points over the closest system, but still significant on the ja-en task, where the use of source-side syntax has proven less effective in previous work (Sudoh et al., 2011). The next thing to notice is that if we had instead used T2S-all, our conclusion would have been much different. This system is able to achieve respectable accuracy compared to PBMT or Hiero, but does not exceed the more competitive pre/post-ordering systems.[3] With this result in hand, we will investigate the contribution of each of these elements in detail in the following sections. In the remainder of the paper settings follow T2S+all except when otherwise noted.

3 Parsing

3.1 Parsing Overview

As T2S translation uses parse trees both in training and testing of the system, an accurate syntactic parser is required. In order to test the extent that parsing accuracy affects translation, we use two

[1] Stanford/Eda, GIZA++, pop-limit 5000 cube pruning.

[2] Egret forests, Nile, pop-limit 5000 hypergraph search.

[3] We have also observed similar trends on other genres and language pairs. For example, in a Japanese-Chinese/English medical conversation task (Neubig et al., 2013), forests, alignment, and search resulted in BLEU increases of en-ja 24.55→30.81, ja-en 19.28→22.46, zh-ja 15.22→20.67, ja-zh 30.88→33.89.

different syntactic parsers and examine the translation accuracy realized by each parser.

For English, the two most widely referenced parsers are the Stanford Parser and Berkeley Parser. In this work, we compare the Stanford Parser's CFG model, with the Berkeley Parser's latent variable model. In previous reports, it has been noted (Kummerfeld et al., 2012) that the latent variable model of the Berkeley parser tends to have the higher accuracy of the two, so if the accuracy of a system using this model is higher then it is likely that parsing accuracy is important for T2S translation. Instead of the Berkeley Parser itself, we use a clone Egret,[4] which achieves nearly identical accuracy, and is able to output packed forests for use in MT, as mentioned below. Trees are right-binarized, with the exception of phrase-final punctuation, which is split off before any other element in the phrase.

For Japanese, our first method uses the MST-based pointwise dependency parser of Flannery et al. (2011), as implemented in the Eda toolkit.[5] In order to convert dependencies into phrase-structure trees typically used in T2S translation, we use the head rules implemented in the Travatar toolkit. In addition, we also train a latent variable CFG using the Berkeley Parser and use Egret for parsing. Both models are trained on the Japanese Word Dependency Treebank (Mori et al., 2014).

In addition, Mi et al. (2008) have proposed a method for forest-to-string (F2S) translation using packed forests to encode many possible sentence interpretations. By doing so, it is possible to resolve some of the ambiguity in syntactic interpretation at translation time, potentially increasing translation accuracy. However, the great majority of recent works on T2S translation do not consider multiple syntactic parses (e.g. Liu et al. (2011), Zhang et al. (2011)), and thus it is important to confirm the potential gains that could be acquired by taking ambiguity into account.

3.2 Effect of Parsing and Forest Input

In Table 2 we show the results for Stanford/Eda with 1-best tree input vs. Egret with trees or forests as input. Forests are those containing all edges in the 100-best parses.

	en-ja		ja-en	
System	BLEU	RIBES	BLEU	RIBES
Stan/Eda	38.95	78.47	32.56	73.03
Egret-T	39.26	79.26	32.97	74.94
Egret-F	**40.84**	**80.15**	**33.70**	**75.94**

Table 2: Results for Stanford/Eda, Egret with tree input, and Egret with forest input.

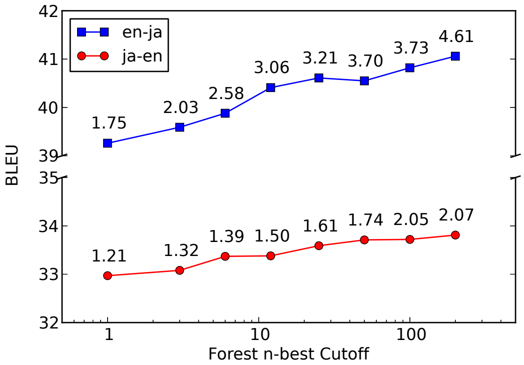

Figure 1: BLEU scores using various levels of forest pruning. Numbers in the graph indicate decoding time in seconds/sentence.

First looking at the difference between the two parsers, we can see that the T2S system using Egret achieves greater accuracy than that using the other two parsers. This improvement is particularly obvious in RIBES, indicating that an increase in parsing accuracy has a larger effect on global reordering than on lexical choice. When going from T2S to F2S translation using Egret, we see another large gain in accuracy, although this time with the gain in BLEU being more prominent. We believe this is related to the observation of Zhang and Chiang (2012) that F2S translation is not necessarily helping fixing parsing errors, but instead giving the translation system the freedom to ignore the parse somewhat, allowing for less syntactically motivated but more fluent translations.

As passing some degree of syntactic ambiguity on to the decoder through F2S translation has proven useful, a next natural question is how much of this ambiguity we need to preserve in our forest. The pruning criterion that we use for the forest is based on including all edges that appear in one or more of the *n*-best parses, so we perform translation setting *n* to 1 (trees), 3, 6, 12, 25, 50, 100, and 200. Figure 1 shows results for these settings with regards to translation accuracy and speed. Overall, we can see that every time we double the size of the forest we get an approximately linear in-

[4] http://code.google.com/p/egret-parser

[5] http://plata.ar.media.kyoto-u.ac.jp/tool/EDA

crease in BLEU at the cost of an increase in decoding time. Interestingly, the increases in BLEU did not show any sign of saturating even when setting the *n*-best cutoff to 200, although larger cutoffs resulted in exceedingly large translation forests that required large amounts of memory.

4 Alignment

4.1 Alignment Overview

The second element that we investigate is alignment accuracy. It has been noted in many previous works that significant gains in alignment accuracy do not make a significant difference in translation results (Ayan and Dorr, 2006; Ganchev et al., 2008). However, none of these works have explicitly investigated the effect on T2S translation, so it is not clear whether these results carry over to our current situation.

As our baseline aligner, we use the GIZA++ implementation of the IBM models (Och and Ney, 2003) with the default options. To test the effect of improved alignment accuracy, we use the discriminative alignment method of Riesa and Marcu (2010) as implemented in the Nile toolkit.[6] This method has the ability to use source- and target-side syntactic information, and has been shown to improve the accuracy of S2T translation.

We trained Nile and tested both methods on the Japanese-English alignments provided with the Kyoto Free Translation Task (Neubig, 2011) (430k parallel sentences, 1074 manually aligned training sentences, and 120 manually aligned test sentences).[7] As creating manual alignment data is costly, we also created two training sets that consisted of 1/4 and 1/16 of the total data to test if we can achieve an effect with smaller amounts of manually annotated data. The details of data size and alignment accuracy are shown in Table 3.

4.2 Effect of Alignment on Translation

In Table 4, we show results when we vary the aligner between GIZA++ and Nile. For reference, we also demonstrate results when using the same alignments for PBMT and Hiero.

From this, we can see that while for PBMT and Hiero systems the results are mixed, as has been noted in previous work (Fraser and Marcu, 2007), improving the alignment accuracy gives significant gains for T2S translation. The reason for this difference is two-fold. The first is that in rule extraction in syntax-based translation (Galley et al., 2006), a single mistaken alignment crossing phrase boundaries results not only in a bad rule being extracted, but also prevents the extraction of a number of good rules. This is reflected in the size of the rule table; the en-ja system built using Nile contains 92.8M rules, while the GIZA++ system contains only 83.3M rules, a 11.2% drop.

The second reason why alignment is important is that while one of the merits of T2S models is their ability to perform global re-ordering, it is difficult to learn good reorderings from bad alignments. We show an example of this in Figure 2. When translating SVO English to SOV Japanese, we expect rules containing a verb and a following noun phrase (VO) to have a high probability of being reversed (to OV), possibly with the addition of

Name	Sent.	Prec.	Rec.	F-meas
GIZA++	0	60.46	55.48	57.86
Nile/16	68	70.21	60.81	65.17
Nile/4	269	72.85	62.70	67.40
Nile	1074	72.73	63.97	68.07

Table 3: Alignment accuracy (%) by method and number of manually annotated training sentences.

	en-ja		ja-en	
System	BLEU	RIBES	BLEU	RIBES
PBMT-G	35.84	72.89	30.49	69.80
PBMT-N	36.05	71.84	30.77	69.75
Hiero-G	34.45	72.94	29.41	69.51
Hiero-N	33.90	72.63	28.90	69.83
T2S-G	39.57	78.94	32.62	75.19
T2S-N/16	**40.79**	**80.05**	32.82	74.89
T2S-N/4	**40.97**	**80.32**	33.35	**75.46**
T2S-N	**40.84**	**80.15**	**33.70**	**75.94**

Table 4: Results varying the aligner (**G**IZA++ vs. **N**ile), including results for Nile when using 1/4 or 1/16 of the annotated training data.

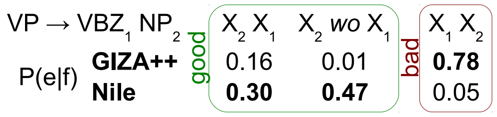

Figure 2: Probabilities for SVO→SOV rules.

[6] http://code.google.com/p/nile

[7] This data is from Wikipedia articles about Kyoto City, and is an entirely different genre than our MT test data. It is likely that creating aligned data that matches the MT genre would provide larger gains in MT accuracy.

the Japanese direct object particle "wo." From the figure, we can see that the probabilities learned by Nile match this intuition, while the probabilities learned by GIZA heavily favor no reordering.

Finally, looking at the amount of data needed to train the model, we can see that a relatively small amount of manually annotated data proves sufficient for large gains in alignment accuracy, with even 68 sentences showing a 7.31 point gain in F-measure over GIZA++. This is because Nile's feature set uses generalizable POS/syntactic information and also because mis-alignments of common function words (e.g. a/the) will be covered even by small sets of training data. Looking at the MT results, we can see that even the smaller data sets allow for gains in accuracy, although the gains are more prominent for en-ja.

5 Search

5.1 Search Overview

Finally, we examine the effect that the choice of search algorithm has on the accuracy of translation. The most standard search algorithm for T2S translation is bottom-up beam search using cube pruning (CP, Chiang (2007)). However, there are a number of other search algorithms that have been proposed for tree-based translation in general (Huang and Chiang, 2007) or T2S systems in particular (Huang and Mi, 2010; Feng et al., 2012). In this work, we compare CP and the hypergraph search (HS) method of Heafield et al. (2013), which is also a bottom-up pruning algorithm but performs more efficient search by grouping together similar language model states.

5.2 Effect of Search

Figure 3 shows BLEU and decoding speed results using HS or CP on T2S and F2S translation, using a variety of pop limits. From this, we can see that HS out-performs CP for both F2S and T2S, especially with smaller pop limits. Comparing the graphs for F2S and T2S translation, it is notable that the shapes of the graphs for the two methods are strikingly similar. This result is somewhat surprising, as the overall search space of F2S is larger and it would be natural for the characteristics of the search algorithm to vary between these two settings. Finally, comparing ja-en and en-ja, search is simpler for the former, a result of the fact that the Japanese sentences contain more words, and thus more LM evaluations per sentence.

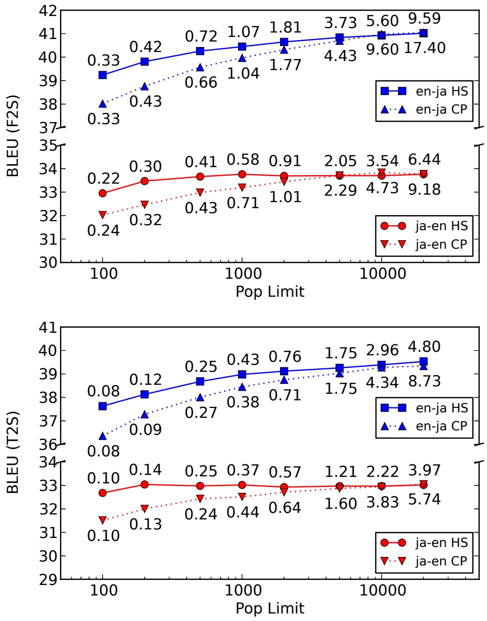

Figure 3: Hypergraph search (HS) and cube pruning (CP) results for F2S and T2S. Numbers above and below the lines indicate time in seconds/sentence for HS and CP respectively.

6 Conclusion

In this paper, we discussed the importance of three peripheral elements that contribute greatly to the accuracy of T2S machine translation: parsing, alignment, and search. Put together, a T2S system that uses the more effective settings for these three elements greatly outperforms a system that uses more standard settings, as well as the current state-of-the-art on English-Japanese and Japanese-English translation tasks.

Based on these results we draw three conclusions. The first is that given the very competitive results presented here, T2S systems do seem to have the potential to achieve high accuracy, even when compared to strong baselines incorporating syntactic reordering into a phrase-based system. The second is that when going forward with research on T2S translation, one should first be sure to account for these three elements to ensure a sturdy foundation for any further improvements. Finally, considering the fact that parsing and alignment for each of these languages is far from perfect, further research investment in these fields may very well have the potential to provide additional gains in accuracy in the T2S framework.

Acknowledgments: This work was supported by JSPS KAKENHI Grant Number 25730136.

References

Vamshi Ambati and Alon Lavie. 2008. Improving syntax driven translation models by re-structuring divergent and non-isomorphic parse tree structures. In *Proc. AMTA*, pages 235–244.

Necip Ayan and Bonnie Dorr. 2006. Going beyond AER: an extensive analysis of word alignments and their impact on MT. In *Proc. ACL*.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. ACL*, pages 531–540.

Yang Feng, Yang Liu, Qun Liu, and Trevor Cohn. 2012. Left-to-right tree-to-string decoding with prediction. In *Proc. EMNLP*, pages 1191–1200.

Daniel Flannery, Yusuke Miyao, Graham Neubig, and Shinsuke Mori. 2011. Training dependency parsers from partially annotated corpora. In *Proc. IJCNLP*, pages 776–784.

Alexander Fraser and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. ACL*, pages 961–968.

Kuzman Ganchev, João V. Graça, and Ben Taskar. 2008. Better alignments = better translations? In *Proc. ACL*.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proceedings of NTCIR*, volume 9, pages 559–578.

Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2012. Post-ordering by parsing for Japanese-English statistical machine translation. In *Proc. ACL*, pages 311–316.

Jonathan Graehl and Kevin Knight. 2004. Training tree transducers. In *Proc. HLT*, pages 105–112.

Kenneth Heafield, Philipp Koehn, and Alon Lavie. 2013. Grouping language model boundary words to speed k–best extraction from hypergraphs. In *Proc. NAACL*, pages 958–968.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proc. ACL*, pages 144–151.

Liang Huang and Haitao Mi. 2010. Efficient incremental decoding for tree-to-string translation. In *Proc. EMNLP*, pages 273–283.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010a. Automatic evaluation of translation quality for distant language pairs. In *Proc. EMNLP*, pages 944–952.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010b. Head finalization: A simple reordering rule for SOV languages. In *Proc. WMT and MetricsMATR*.

Rasoul Samad Zadeh Kaljahi, Raphael Rubino, Johann Roturier, and Jennifer Foster. 2012. A detailed analysis of phrase-based and syntax-based machine translation: The search for systematic differences. In *Proc. AMTA*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, pages 177–180.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP*.

Jonathan K Kummerfeld, David Hall, James R Curran, and Dan Klein. 2012. Parser showdown at the wall street corral: an empirical investigation of error types in parser output. In *Proc. EMNLP*, pages 1048–1059.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proc. ACL*.

Yang Liu, Qun Liu, and Yajuan Lü. 2011. Adjoining tree-to-string translation. In *Proc. ACL*, pages 1278–1287.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proc. ACL*, pages 192–199.

Shinsuke Mori, Hideki Ogura, and Tetsuro Sasada. 2014. A Japanese word dependency corpus. In *Proc. LREC*.

Graham Neubig, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura, Yuji Matsumoto, Ryosuke Isotani, and Yukichi Ikeda. 2013. Towards high-reliability speech translation in the medical domain. In *Proc. MedNLP*, pages 22–29.

Graham Neubig. 2011. The Kyoto free translation task. `http://www.phontron.com/kftt`.

Graham Neubig. 2013. Travatar: A forest-to-string machine translation engine based on tree transducers. In *Proc. ACL Demo Track*, pages 91–96.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318.

Jason Riesa and Daniel Marcu. 2010. Hierarchical search for word alignment. In *Proc. ACL*, pages 157–166.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. ACL*, pages 577–585.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Post-ordering in statistical machine translation. In *Proc. MT Summit*.

Akihiro Tamura, Taro Watanabe, Eiichiro Sumita, Hiroya Takamura, and Manabu Okumura. 2013. Part-of-speech induction in dependency trees for statistical machine translation. In *Proc. ACL*, pages 841–851.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proc. EMNLP*, pages 216–226.

Hui Zhang and David Chiang. 2012. An exploration of forest-to-string translation: Does translation help or hurt parsing? In *Proc. ACL*, pages 317–321.

Hao Zhang, Licheng Fang, Peng Xu, and Xiaoyun Wu. 2011. Binarized forest to string translation. In *Proc. ACL*, pages 835–845.