An Exploration of Forest-to-String Translation: Does Translation Help or Hurt Parsing?

Hui Zhang
University of Southern California
Department of Computer Science
hzhang@isi.edu

David Chiang
University of Southern California
Information Sciences Institute
chiang@isi.edu

Abstract

Syntax-based translation models that operate on the output of a source-language parser have been shown to perform better if allowed to choose from a set of possible parses. In this paper, we investigate whether this is because it allows the translation stage to overcome parser errors or to override the syntactic structure itself. We find that it is primarily the latter, but that under the right conditions, the translation stage does correct parser errors, improving parsing accuracy on the Chinese Treebank.

1 Introduction

Tree-to-string translation systems (Liu et al., 2006; Huang et al., 2006) typically employ a pipeline of two stages: a syntactic parser for the source language, and a decoder that translates source-language trees into target-language strings. Originally, the output of the parser stage was a single parse tree, and this type of system has been shown to outperform phrase-based translation on, for instance, Chinese-to-English translation (Liu et al., 2006). More recent work has shown that translation quality is improved further if the parser outputs a weighted parse *forest*, that is, a representation of a whole distribution over possible parse trees (Mi et al., 2008). In this paper, we investigate two hypotheses to explain why.

One hypothesis is that *forest-to-string translation selects worse parses*. Although syntax often helps translation, there may be situations where syntax, or at least syntax in the way that our models use it, can impose constraints that are too rigid for good-quality translation (Liu et al., 2007; Zhang et al., 2008). For example, suppose that a tree-to-string system encounters the following correct tree (only partial bracketing shown):

(1) [$_{\text{NP}}$ jīngjì zēngzhǎng] de sùdù
economy growth DE rate
'economic growth rate'

Suppose further that the model has never seen this phrase before, although it has seen the subphrase *zēngzhǎng de sùdù* 'growth rate'. Because this subphrase is not a syntactic unit in sentence (1), the system will be unable to translate it. But a forest-to-string system would be free to choose another (incorrect but plausible) bracketing:

(2) jīngjì [$_{\text{NP}}$ zēngzhǎng de sùdù]
economy growth DE rate

and successfully translate it using rules learned from observed data.

The other hypothesis is that *forest-to-string translation selects better parses*. For example, if a Chinese parser is given the input *cānjiā biǎojiě de hūnlǐ*, it might consider two structures:

(3) [$_{\text{VP}}$ cānjiā biǎojiě] de hūnlǐ
attend cousin DE wedding
'wedding that attends a cousin'

(4) cānjiā [$_{\text{NP}}$ biǎojiě de hūnlǐ]
attend cousin DE wedding
'attend a cousin's wedding'

The two structures have two different translations into English, shown above. While the parser prefers structure (3), an *n*-gram language model would easily prefer translation (4) and, therefore, its corresponding Chinese parse.

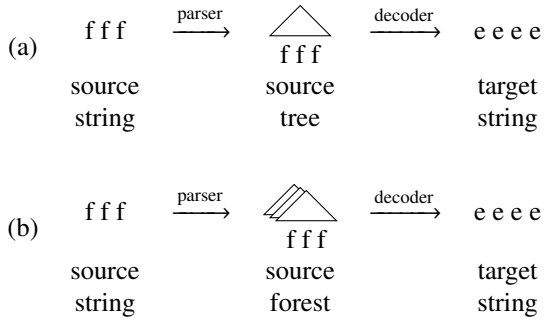

Figure 1: (a) In tree-to-string translation, the parser generates a single tree which the decoder must use to generate a translation. (b) In forest-to-string translation, the parser generates a forest of possible trees, any of which the decoder can use to generate a translation.

Previous work has shown that an observed target-language translation can improve parsing of source-language text (Burkett and Klein, 2008; Huang et al., 2009), but to our knowledge, only Chen et al. (2011) have explored the case where the target-language translation is unobserved.

Below, we carry out experiments to test these two hypotheses. We measure the accuracy (using labeled-bracket F1) of the parses that the translation model selects, and find that they are worse than the parses selected by the parser. Our basic conclusion, then, is that the parses that help translation (according to BLEU) are, on average, worse parses. That is, forest-to-string translation hurts parsing.

But there is a twist. Neither labeled-bracket F1 nor BLEU is a perfect metric of the phenomena it is meant to measure, and our translation system is optimized to maximize BLEU. If we optimize our system to maximize labeled-bracket F1 instead, we find that our translation system selects parses that score higher than the baseline parser's. That is, forest-to-string translation can help parsing.

2 Background

We provide here only a cursory overview of tree-to-string and forest-to-string translation. For more details, the reader is referred to the original papers describing them (Liu et al., 2006; Mi et al., 2008).

Figure 1a illustrates the tree-to-string translation pipeline. The parser stage can be any phrase-structure parser; it computes a parse for each source-language string. The decoder stage translates the source-language tree into a target-language string, using a synchronous tree-substitution grammar.

In forest-to-string translation (Figure 1b), the parser outputs a forest of possible parses of each source-language string. The decoder uses the same rules as in tree-to-string translation, but is free to select any of the trees contained in the parse forest.

3 Translation hurts parsing

The simplest experiment to carry out is to examine the parses actually selected by the decoder, and see whether they are better or worse than the parses selected by the parser. If they are worse, this supports the hypothesis that syntax can hurt translation. If they are better, we can conclude that translation can help parsing. In this initial experiment, we find that the former is the case.

3.1 Setup

The baseline parser is the Charniak parser (Charniak, 2000). We trained it on the Chinese Treebank (CTB) 5.1, split as shown in Table 1, following Duan et al. (2007).[1] The parser outputs a parse forest annotated with head words and other information. Since the decoder does not use these annotations, we use the max-rule algorithm (Petrov et al., 2006) to (approximately) sum them out. As a side benefit, this improves parsing accuracy from 77.76% to 78.42% F1. The weight of a hyperedge in this forest is its posterior probability, given the input string. We retain these weights as a feature in the translation model.

The decoder stage is a forest-to-string system (Liu et al., 2006; Mi et al., 2008) for Chinese-to-English translation. The datasets used are listed in Table 1. We generated word alignments with GIZA++ and symmetrized them using the *grow-diag-final-and* heuristic. We parsed the Chinese side using the Charniak parser as described above, and performed forest-based rule extraction (Mi and Huang, 2008) with a maximum height of 3 nodes. We used the same features as Mi and Huang (2008). The language model was a trigram model with modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998), trained on the target

[1] The more common split, used by Bikel and Chiang (2000), has flaws that are described by Levy and Manning (2003).

	Parsing	Translation
Train	CTB 1–815 CTB 1101–1136	FBIS
Dev	CTB 900–931 CTB 1148–1151	NIST 2002
Test	CTB 816–885 CTB 1137–1147	NIST 2003

Table 1: Data used for training and testing the parsing and translation models.

System	Objective	Parsing F1%	Translation BLEU%
Charniak	n/a	78.42	n/a
tree-to-string	max-BLEU	78.42	23.07
forest-to-string	max-BLEU	77.75	24.60
forest-to-string	max-F1	78.81	19.18

Table 2: Forest-to-string translation outperforms tree-to-string translation according to BLEU, but the decreases parsing accuracy according to labeled-bracket F1. However, when we train to maximize labeled-bracket F1, forest-to-string translation yields better parses than both tree-to-string translation and the original parser.

side of the training data. We used minimum-error-rate (MER) training to optimize the feature weights (Och, 2003) to maximize BLEU.

At decoding time, we select the best derivation and extract its source tree. In principle, we ought to sum over all derivations for each source tree; but the approximations that we tried (*n*-best list crunching, max-rule decoding, minimum Bayes risk) did not appear to help.

3.2 Results

Table 2 shows the main results of our experiments. In the second and third line, we see that the forest-to-string system outperforms the tree-to-string system by 1.53 BLEU, consistent with previously published results (Mi et al., 2008; Zhang et al., 2009). However, we also find that the trees selected by the forest-to-string system score much lower according to labeled-bracket F1. This suggests that the reason the forest-to-string system is able to generate better translations is that it can soften the constraints imposed by the syntax of the source language.

4 Translation helps parsing

We have found that better translations can be obtained by settling for worse parses. However, translation accuracy is measured using BLEU and parsing accuracy is measured using labeled-bracket F1, and neither of these is a perfect metric of the phenomenon it is meant to measure. Moreover, we optimized the translation model in order to maximize BLEU. It is known that when MER training is used to optimize one translation metric, other translation metrics suffer (Och, 2003); much more, then, can we expect that optimizing BLEU will cause labeled-bracket F1 to suffer. In this section, we try optimizing labeled-bracket F1, and find that, in this case, the translation model does indeed select parses that are better on average.

4.1 Setup

MER training with labeled-bracket F1 as an objective function is straightforward. At each iteration of MER training, we run the parser and decoder over the CTB dev set to generate an *n*-best list of possible translation derivations (Huang and Chiang, 2005). For each derivation, we extract its Chinese parse tree and compute the number of brackets guessed and the number matched against the gold-standard parse tree. A trivial modification of the MER trainer then optimizes the feature weights to maximize labeled-bracket F1.

A technical challenge that arises is ensuring diversity in the *n*-best lists. The MER trainer requires that each list contain enough unique translations (when maximizing BLEU) or source trees (when maximizing labeled-bracket F1). However, because one source tree may lead to many translation derivations, the *n*-best list may contain only a few unique source trees, or in the extreme case, the derivations may all have the same source tree. We use a variant of the *n*-best algorithm that allows efficient generation of equivalence classes of derivations (Huang et al., 2006). The standard algorithm works by generating, at each node of the forest, a list of the best subderivations at that node; the variant drops a subderivation if it has the same source tree as a higher-scoring subderivation.

Maximum rule height	F1%
3	78.81
4	78.93
5	79.14

(a)

LM data (lines)	F1%
none	78.78
100	78.79
30k	78.67
300k	79.14
13M	79.24

(b)

Features	F1%
monolingual	78.89
+ bilingual	79.24

(c)

Parallel data (lines)	F1%
60k	78.00
120k	78.16
300k	79.24

(d)

Table 3: Effect of variations on parsing performance. (a) Increasing the maximum translation rule height increases parsing accuracy further. (b) Increasing/decreasing the language model size increases/decreases parsing accuracy. (c) Decreasing the parallel text size decreases parsing accuracy. (d) Removing all bilingual features decreases parsing accuracy, but only slightly.

4.2 Results

The last line of Table 2 shows the results of this second experiment. The system trained to optimize labeled-bracket F1 (*max-F1*) obtains a much lower BLEU score than the one trained to maximize BLEU (*max*-BLEU)—unsurprisingly, because a single source-side parse can yield many different translations, but the objective function scores them equally. What is more interesting is that the max-F1 system obtains a higher F1 score, not only compared with the max-BLEU system but also the original parser.

We then tried various settings to investigate what factors affect parsing performance. First, we found that increasing the maximum rule height increases F1 further (Table 3a).

One of the motivations of our method is that bilingual information (especially the language model) can help disambiguate the source side structures. To test this, we varied the size of the corpus used to train the language model (keeping a maximum rule height of 5 from the previous experiment). The 13M-line language model adds the Xinhua portion of Gigaword 3. In Table 3b we see that the parsing performance does increase with the language model size, with the largest language model yielding a net improvement of 0.82 over the baseline parser.

To test further the importance of bilingual information, we compared against a system built only from the Chinese side of the parallel text (with each word aligned to itself). We removed all features that use bilingual information, retaining only the parser probability and the phrase penalty. In their place we added a new feature, the probability of a rule's source side tree given its root label, which is essentially the same model used in Data-Oriented Parsing (Bod, 1992). Table 3c shows that this system still outperforms the original parser. In other words, part of the gain is not attributable to translation, but additional source-side context and data that the translation model happens to capture.

Finally, we varied the size of the parallel text (keeping a maximum rule height of 5 and the largest language model) and found that, as expected, parsing performance correlates with parallel data size (Table 3d).

5 Conclusion

We set out to investigate why forest-to-string translation outperforms tree-to-string translation. By comparing their performance as Chinese parsers, we found that forest-to-string translation sacrifices parsing accuracy, suggesting that forest-to-string translation works by overriding constraints imposed by syntax. But when we optimized the system to maximize labeled-bracket F1, we found that, in fact, forest-to-string translation is able to achieve higher accuracy, by 0.82 F1%, than the baseline Chinese parser, demonstrating that, to a certain extent, forest-to-string translation is able to correct parsing errors.

Acknowledgements

We are grateful to the anonymous reviewers for their helpful comments. This research was supported in part by DARPA under contract DOI-NBC D11AP00244.

References

Daniel M. Bikel and David Chiang. 2000. Two statistical parsing models applied to the Chinese Treebank. In *Proc. Second Chinese Language Processing Workshop*, pages 1–6.

Rens Bod. 1992. A computational model of language performance: Data Oriented Parsing. In *Proc. COLING 1992*, pages 855–859.

David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proc. EMNLP 2008*, pages 877–886.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. NAACL*, pages 132–139.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University Center for Research in Computing Technology.

Wenliang Chen, Jun'ichi Kazama, Min Zhang, Yoshimasa Tsuruoka, Yujie Zhang, Yiou Wang, Kentaro Torisawa, and Haizhou Li. 2011. SMT helps bitext dependency parsing. In *Proc. EMNLP 2011*, pages 73–83.

Xiangyu Duan, Jun Zhao, and Bo Xu. 2007. Probabilistic models for action-based Chinese dependency parsing. In *Proc. ECML 2007*, pages 559–566.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proc. IWPT 2005*, pages 53–64.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proc. AMTA 2006*, pages 65–73.

Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proc. EMNLP 2009*, pages 1222–1231.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for M-gram language modeling. In *Proc. ICASSP 1995*, pages 181–184.

Roger Levy and Christopher D. Manning. 2003. Is it harder to parse Chinese, or the Chinese Treebank? In *Proc. ACL 2003*, pages 439–446.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proc. COLING-ACL 2006*, pages 609–616.

Yang Liu, Yun Huang, Qun Liu, and Shouxun Lin. 2007. Forest-to-string statistical translation rules. In *Proc. ACL 2007*, pages 704–711.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proc. EMNLP 2008*, pages 206–214.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proc. ACL-08: HLT*, pages 192–199.

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

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. COLING-ACL 2006*, pages 433–440.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proc. ACL-08: HLT*, pages 559–567.

Hui Zhang, Min Zhang, Haizhou Li, Aiti Aw, and Chew Lim Tan. 2009. Forest-based tree sequence to string translation model. In *Proc. ACL-IJCNLP 2009*, pages 172–180.