Sub-Sentence Division for Tree-Based Machine Translation

Hao Xiong[*], **Wenwen Xu**[+], **Haitao Mi**[*], **Yang Liu**[*] and **Qun Liu**[*]
[*]Key Lab. of Intelligent Information Processing
[+]Key Lab. of Computer System and Architecture
Institute of Computing Technology
Chinese Academy of Sciences
P.O. Box 2704, Beijing 100190, China
{xionghao,xuwenwen,htmi,yliu,liuqun}@ict.ac.cn

Abstract

Tree-based statistical machine translation models have made significant progress in recent years, especially when replacing 1-best trees with packed forests. However, as the parsing accuracy usually goes down dramatically with the increase of sentence length, translating long sentences often takes long time and only produces degenerate translations. We propose a new method named sub-sentence division that reduces the decoding time and improves the translation quality for tree-based translation. Our approach divides long sentences into several sub-sentences by exploiting tree structures. Large-scale experiments on the NIST 2008 Chinese-to-English test set show that our approach achieves an absolute improvement of 1.1 BLEU points over the baseline system in 50% less time.

1 Introduction

Tree-based statistical machine translation models in days have witness promising progress in recent years, such as tree-to-string models (Liu et al., 2006; Huang et al., 2006), tree-to-tree models (Quirk et al.,2005;Zhang et al., 2008). Especially, when incorporated with forest, the correspondent forest-based tree-to-string models (Mi et al., 2008; Zhang et al., 2009), tree-to-tree models (Liu et al., 2009) have achieved a promising improvements over correspondent tree-based systems. However, when we translate long sentences, we argue that two major issues will be raised. On one hand, parsing accuracy will be lower as the length of sentence grows. It will inevitably hurt the translation quality (Quirk and Corston-Oliver, 2006; Mi and Huang, 2008). On the other hand, decoding on long sentences will be time consuming, especially for forest approaches. So splitting long sentences into sub-sentences becomes a natural way in MT literature.

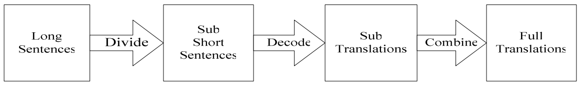

Figure 1. Main framework of our method

A simple way is to split long sentences by punctuations. However, without concerning about the original whole tree structures, this approach will result in ill-formed sub-trees which don't respect to original structures. In this paper, we present a new approach, which pays more attention to parse trees on the long sentences. We firstly parse the long sentences into trees, and then divide them accordingly into sub-sentences, which will be translated independently (Section 3). Finally, we combine sub translations into a full translation (Section 4). Large-scale experiments (Section 5) show that the BLEU score achieved by our approach is 1.1 higher than direct decoding and 0.3 higher than always splitting on commas on the 2008 NIST MT Chinese-English test set. Moreover, our approach has reduced decoding time significantly.

2 Framework

Our approach works in following steps.
(1) Split a long sentence into sub-sentences.
(2) Translate all the sub-sentences respectively.
(3) Combine the sub-translations.

Figure 1 illustrates the main idea of our approach. The crucial issues of our method are how to divide long sentences and how to combine the sub-translations.

3 Sub Sentence Division

Long sentences could be very complicated in grammar and sentence structure, thereby creating an obstacle for translation. Consequently, we need to break them into shorter and easier clauses. To divide sentences by punctuation is

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 137–140,
Suntec, Singapore, 4 August 2009.

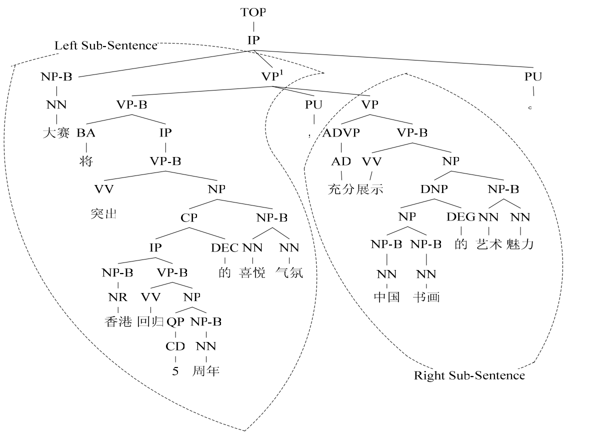

Figure 2. An undividable parse tree

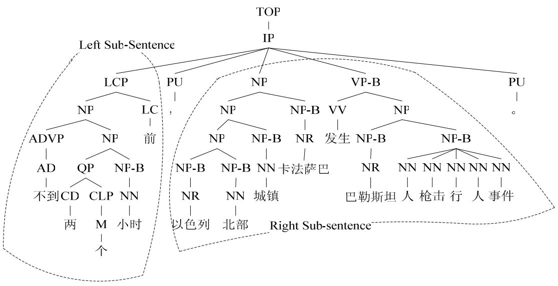

Figure 3. A dividable parse tree

one of the most commonly used methods. However, simply applying this method might damage the accuracy of parsing. As a result, the strategy we proposed is to operate division while concerning the structure of parse tree.

As sentence division should not influence the accuracy of parsing, we have to be very cautious about sentences whose division might decrease the accuracy of parsing. Figure 2(a) shows an example of the parse tree of an undividable sentence.

As can be seen in Figure 2, when we divide the sentence by comma, it would break the structure of "VP" sub-tree and result in a ill-formed sub-tree "VP" (right sub-tree), which don't have a subject and don't respect to original tree structures.

Consequently, the key issue of sentence division is finding the sentences that can be divided without loosing parsing accuracy. Figure 2(b) shows the parse tree of a sentence that can be divided by punctuation, as sub-sentences divided by comma are independent. The reference translation of the sentence in figure 3 is

Less than two hours earlier, a Palestinian took on a shooting spree on passengers in the town of Kfar Saba in northern Israel.

Pseudocode 1 Check Sub Sentence Division Algorithm

```
procedure CheckSubSentence(sent)
for each word i in sent
  if(i is a comma)
    left={words in left side of i};
     //words between last comma and current comma i
    right={words in right side of i};
     //words between i and next comma or semicolon, period, question mark
    isDividePunct[i]=true;
    for each j in left
      if(( LCA(j, i)!=parent[i])
        isDividePunct[i]=false;
        break;
    for each j in right
      if(( LCA(j, i)!=parent[i])
        isDividePunct[i]=false;
        break;
function LCA(i, j)
  return lowest common ancestor(i, j);
```

It demonstrates that this long sentence can be divided into two sub-sentences, providing a good support to our division.

In addition to dividable sentences and non-dividable sentences, there are sentences containing more than one comma, some of which are dividable and some are not. However, this does not prove to be a problem, as we process each comma independently. In other words, we only split the dividable part of this kind of sentences, leaving the non-dividable part unchanged.

To find the sentences that can be divided, we present a new method and provide its pseudo code. Firstly, we divide a sentence by its commas. For each word in the sub-sentence on the left side of a comma, we compute its lowest common ancestor (LCA) with the comma. And we process the words in the sub-sentence on the right side of the comma in the same way. Finally, we check if all the LCA we have computed are comma's parent node. If all the LCA are the comma's parent node, the sub-sentences are independent.

As shown in figure 3, the LCA (AD 不到 , PU ，), is "IP" ,which is the parent node of "PU ， "; and the LCA (NR 以色列 , PU ，) is also "IP". Till we have checked all the LCA of each word and comma, we finally find that all the LCA are "IP". As a result, this sentence can be divided without loosing parsing accuracy. LCA can be computed by using union-set (Tarjan, 1971) in lineal time. Concerning the

sub-sentence 1: 强卓指出	
Translation 1: Johndroe said	*A1*
Translation 2: Johndroe pointed out	*A2*
Translation 3: Qiang Zhuo said	*A3*
comma 1: , ***Translation***: punctuation translation (white space, that …)	
sub-sentence 2: 两位总统也对昨日签署的美国—南韩自由贸易协议表示欢迎	
Translation 1: the two presidents also welcomed the US-South Korea free trade agreement that was signed yesterday	*B1*
Translation 2: the two presidents also expressed welcome to the US – South Korea free trade agreement signed yesterday	*B2*
comma 2: , ***Translation***: punctuation translation (white space, that …)	
sub-sentence 3:并将致力确保两国国会批准此一协议。	
Translation 1: and would work to ensure that the congresses of both countries approve this agreement.	*C1*
Translation 2: and will make efforts to ensure the Congress to approve this agreement of the two countries.	*C2*

Table 1. Sub translation example

implementation complexity, we have reduced the problem to range minimum query problem (Bender et al., 2005) with a time complexity of $O(1)$ for querying.

Above all, our approach for sub sentence works as follows:

(1)Split a sentence by semi-colon if there is one.
(2)Parse a sentence if it contains a comma, generating k-best parses (Huang Chiang, 2005) with k=10.
(3)Use the algorithm in pseudocode 1 to check the sentence and divide it if there are more than 5 parse trees indicates that the sentence is dividable.

4 Sub Translation Combining

For sub translation combining, we mainly use the best-first expansion idea from *cube pruning* (Huang and Chiang, 2007) to combine sub-translations and generate the whole *k*-best translations. We first select the best translation from sub translation sets, and then use an interpolation language model for rescoring (Huang and Chiang, 2007).

Test Set	02	05	08
No Sent Division	34.56	31.26	24.53
Split by Comma	34.59	31.23	25.39
Our Approach	34.86	31.23	25.69

Table 2. BLEU results (case sensitive)

Test Set	02	05	08
No Sent Division	28 h	36 h	52 h
Split by Comma	18h	23h	29h
Our Approach	18 h	22 h	26 h

Table 3. Decoding time of our experiments
(h means hours)

For example, we split the following sentence "*强卓指出,两位总统也对昨日签署的美国—南韩自由贸易协议表示欢迎,并将致力确保两国国会批准此一协议。*" into three sub-sentences and generate some translations, and the results are displayed in Table 1.

As seen in Table 1, for each sub-sentence, there are one or more versions of translation. For convenience, we label the three translation versions of sub-sentence 1 as A1, A2, and A3, respectively. Similarly, B1, B2, C1, C2 are also labels of translation. We push the A1, white space, B1, white space, C1 into the cube, and then generate the final translation.

According to cube pruning algorithm, we will generate other translations until we get the best list we need. Finally, we rescore the k-best list using interpolation language model and find the best translation which is *A1 that B1 white space C1*.

5 Experiments

5.1 Data preparation

We conduct our experiments on Chinese-English translation, and use the Chinese parser of Xiong et al. (2005) to parse the source sentences. And our decoder is based on forest-based tree-to-string translation model (Mi et al. 2008).

Our training corpus consists of 2.56 million sentence pairs. Forest-based rule extractor (Mi and Huang 2008) is used with a pruning threshold p=3. And we use SRI Language Modeling Toolkit (Stolcke, 2002) to train two 5-gram language models with Kneser-Ney smoothing on the English side of the training corpus and the Xinhua portion of Gigaword corpora respectively.

We use 2006 NIST MT Evaluation test set as development set, and 2002, 2005 and 2008 NIST MT Evaluation test sets as test sets. We also use *minimum error-rate training* (Och, 2003) to tune our feature weights. We evaluate our results with *case-sensitive* BLEU-4 metric (Papineni et al., 2002). The pruning threshold *p* for parse forest in decoding time is 12.

5.2 Results

The final BLEU results are shown in Table 2, our approach has achieved a BLEU score that is 1.1 higher than direct decoding and 0.3 higher than always splitting on commas.

The decoding time results are presented in Table 3. The search space of our experiment is extremely large due to the large pruning threshold (p=12), thus resulting in a long decoding time. However, our approach has reduced the decoding time by 50% over direct decoding, and 10% over always splitting on commas.

6 Conclusion & Future Work

We have presented a new sub-sentence division method and achieved some good results. In the future, we will extend our work from decoding to training time, where we divide the bilingual sentences accordingly.

Acknowledgement

The authors were supported by National Natural Science Foundation of China, Contracts 0873167 and 60736014, and 863 State Key Project No.2006AA010108. We thank Liang Huang for his insightful suggestions.

References

Bender, Farach-Colton, Pemmasani, Skiena, Sumazin, *Lowest common ancestors in trees and directed acyclic graphs.* J. Algorithms 57(2), 75–94 (2005)

Liang Huang and David Chiang. 2005. *Better kbest Parsing.* In *Proceedings of IWPT-2005*.

Liang Huang and David Chiang. 2007. *Forest rescoring: Fast decoding with integrated language models.* In *Proceedings of ACL*.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. *Statistical syntax-directed translation with extended domain of locality.* In *Proceedings of AMTA*

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. *Statistical phrase-based translation.* In *Proceedings of HLT-NAACL 2003*, pages 127-133.

Yang Liu, Qun Liu and Shouxun Lin. 2006. *Tree-to-String alignments template for statistical machine translation.* In *Proceedings of ACL*.

Yang Liu, Yajuan Lv and Qun Liu.2009. *Improving Tree-to-Tree Translation with Packed Forests.*To appear in *Proceedings of ACL/IJCNLP*..

Daniel Marcu, Wei Wang, AbdessamadEchihabi, and Kevin Knight. 2006. *Statistical Machine Translation with syntactifiedtarget language phrases.* In *Proceedings of EMNLP*.

Haitao Mi, Liang Huang, and Qun Liu. 2008. *Forest-based translation.* In *Proceedings of ACL: HLT.*

Haitao Mi and Liang Huang. 2008. *Forest-based translation rule extraction.* In *Proceedings of EMNLP*.

Franz J. Och. 2003. *Minimum error rate training in statistical machine translation.* In *Proceedings of 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 *Proceedings of ACL, pages 311–318,*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. *Dependency treelet translation: Syntactically informed phrasal SMT.* In *Proceedings of ACL.*

Chris Quirk and Simon Corston-Oliver. 2006. *The impact of parse quality on syntactically-informed statistical machine translation.* In *Proceedings of EMNLP*.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings* of *ICSLP,* volume 30, pages 901–904.

Georgianna Tarjan, *Depth First Search and Linear Graph Algorithms.* SIAM J. Comp. 1:2, pp. 146–160, 1972.

Deyi Xiong, Shuanglong Li, Qun Liu, and Shouxun Lin.2005. *Parsing the Penn Chinese Treebank with semantic knowledge.* In *Proceedings of IJCNLP*.

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 *Proceedings of ACL*.

Hui Zhang, Min Zhang, Haizhou Li, Aiti Aw and Chew Lim Tan. 2009. *Forest-based Tree Sequence to String Translation Model*. To appear in *Proceedings of ACL/IJCNLP*