The forest-based Tree Sequence to String SMT System for CWMT-2009

Hui Zhang[*+], Huashen Liang[^1], Min Zhang[*], Haizhou Li[*] and Chew Lim Tan[+]

[*]Institute for Infocomm Research, Singapore
zhangh1982@gmail.com, mzhang@i2r.a-star.edu.sg, hli@i2r.a-star.edu.sg

[+]National University of Singapore, Singapore
tancl@comp.nus.edu.sg

[^]Harbin Institute of Technology, China
huashen.liang@gmail.com

Abstract: This paper reports I^{2}R's SMT System for CWMT-2009 MT evaluation. In the CWMT-2009 MT evaluation, we use our forest-based tree sequence to string translation system to participate in the Chinese-English single system evaluation track. In this paper, we give an overall introduction of our translation system and then report the experiment details including how we pre-process the training data, system configuration and post-processing procedures.

Keywords: Statistical Machine Translation, syntax, packed forest, tree sequence

I^{2}R 的机器翻译系统技术报告

摘要: 本文是 I^{2}R 参加 CWMT-2009 评测的机器翻译系统技术报告。 在本次评测中，我们采用基于压缩森林的树序列到串的翻译系统参加汉英单系统评测。 本文首先对我们的系统进行了总体的介绍，然后详细描地描述实验细节，包括数据预处理，系统配置和后处理细节等等。

关键词: 统计机器翻译, 句法, 压缩森林, 树序列

1 Introduction

The I^{2}R's CWMT2009 MT system is a fully linguistically-motivated syntax-based statistical MT system. It is featured by two state-of-the-art concepts in SMT research community: packed forest and tree sequence. Recently syntax-based statistical machine translation (SMT) methods have achieved very

[1] HuanShen is an intern student at I^{2}R when doing this work.

promising results and attracted more and more interests in the SMT research community. Fundamentally, syntax-based SMT views translation as a structural transformation process. Therefore, structure divergence and parse errors are two of the major issues that may largely compromise the performance of syntax-based SMT (Zhang et al., 2008; Mi et al., 2008). Many solutions have been proposed to address the above two issues. Among these advances, forest-based modeling (Mi et al., 2008; Mi and Huang, 2008) and tree sequence-based modeling (Liu et al., 2007; Zhang et al., 2008) are two interesting modeling methods with promising results reported. Forest-based modeling aims to improve translation accuracy through digging the potential better parses from *n*-bests (i.e. forest) while tree sequence-based modeling aims to model non-syntactic translations with structured syntactic knowledge. We propose a forest-based tree sequence to string translation method, which is designed to integrate the strengths of the forest-based and the tree sequence-based translation methods.

The remainder of the paper is organized as follows. Section 2 describes the general model while in section 3 and section 4, the key rule extraction and decoding algorithms are elaborated. Experimental details are reported in section 5 and the paper is concluded in section 6.

2 Forest-based tree sequence to string model

In this section, we first explain what a packed forest is and then define the concept of the tree sequence in the context of forest followed by the discussion on our proposed model.

2.1 Packed Forest

A packed forest (forest in short) is a special kind of hyper-graph (Klein and Manning, 2001; Huang and Chiang, 2005), which is used to represent all derivations (i.e. parse trees) for a given sentence under a context free grammar (CFG). A forest F is defined as a triple $< V, E, S >$, where V is non-terminal node set, E is hyper-edge set and S is leaf node set (i.e. all sentence words). A forest F satisfies the following two conditions:

1) Each node n in V should cover a phrase, which is a continuous word sub-sequence in S.

2) Each hyper-edge e in E is defined as $v_f \Rightarrow v_1 \dots v_i \dots v_n, \left(v_i \in (V \cup S), v_f \in V\right)$, where $v_1 \dots v_i \dots v_n$ covers a sequence of continuous and non-overlap phrases, v_f is the father node of the children sequence $v_1 \dots v_i \dots v_n$. The phrase covered by v_f is just the sum of all the phrases covered by each child node v_i.

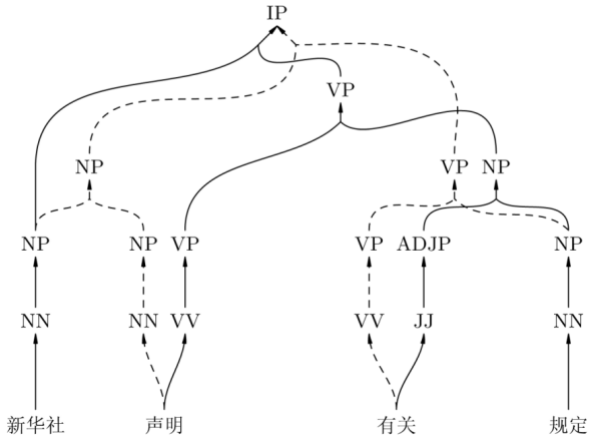

Figure 1. A packed forest for sentence “新华社/Xinhuashe 声明/shengming 有关/youguan 规定/guiding”

We here introduce another concept that is used in our subsequent discussions. A complete forest *CF* is a general forest with one additional condition that there is only one root node *N* in *CF*, i.e., all nodes except the root *N* in a *CF* must have at least one father node.

Fig. 1 is a complete forest while Fig. 3 is a non-complete forest due to the virtual node “VV+VV” introduced in Fig. 3.

2.2 Tree sequence in packed forest

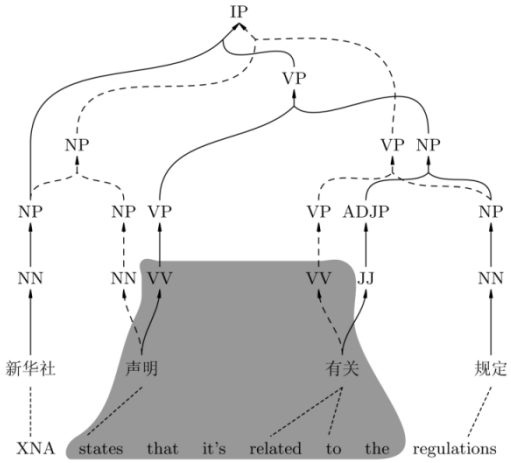

Figure 2. A tree sequence to string rule

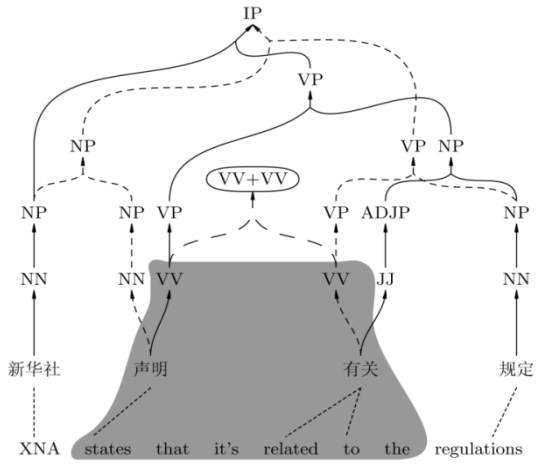

Figure 3. A virtual node in forest

A tree-sequence to string translation rule in a forest is a triple <*L*, *R*, *A*>, where *L* is the tree sequence in source language, *R* is the string containing words and variables in target language, and *A* is the alignment between the leaf nodes of *L* and *R*. This definition is similar to that of (Liu et al. 2007, Zhang et al. 2008) except our tree-sequence is defined in forest. The shaded area of Fig. 2 exemplifies a tree sequence to string translation rule in the forest.

2.3 Forest-based tree-sequence to string translation model

Given a source forest *F* and target translation T_S as well as word alignment A, our translation model is formulated as:

$$\Pr(F, T_S, A) = \sum_{\theta_i \in \Theta, C(\Theta)=(F,T_S,A)} \prod_{r_i \in \theta_i} p(r_i)$$

In the above equation, Θ is the derivation space and θ_i represent a possible derivation path.

By the above Eq., translation becomes a tree sequence structure to string mapping issue. Given the *F*, T_S and A, there are multiple derivations that could map *F* to T_S under the constraint A and we use $C(\Theta) = (F, T_S, A)$ to represent theses multiple derivations.

Our model is implemented under log-linear framework (Och and Ney, 2002). We use seven basic features that are analogous to the commonly used features in phrase-based systems (Koehn, 2003): 1) bidirectional rule mapping probabilities, 2) bidirectional lexical rule translation probabilities, 3) target language model, 4) number of rules used and 5) number of target words. In addition, we define two new features: 1) number of leaf nodes in auxiliary rules (the auxiliary rule will be explained later in this paper) and 2) product of the probabilities of all hyper-edges of the tree sequences in forest.

3 Training

The tree sequence rules can be extracted from a forest in the following two steps:

1) Convert the complete parse forest *F* into a non-complete forest *F* in order to cover those tree sequences that cannot be covered by a single tree node.

2) Employ the forest-based tree rule extraction algorithm (Mi and Huang, 2008) to extract our rules from the non-complete forest.

Theoretically, there is exponential number of node sequences in a forest. To make it manageable, we prune it with the following thresholds:

- each node sequence should contain less than *n* nodes
- each node sequence set should contain less than *m* node sequences
- sort node sequences according to their lengths and only keep the *k* shortest ones

Each virtual node is simply labeled by the concatenation of all its children's labels as shown in Fig. 3.

The first step outputs a non-complete forest *CF* with each alignable span covered by either tree nodes or virtual nodes. Then we can easily extract our rules from the *CF* using the tree rule extraction algorithm (Mi and Huang, 2008).

4 Decoding

We benefit from the same strategy as used in our rule extraction algorithm in designing our decoding algorithm, recasting the forest-based tree sequence-to-string decoding problem as a forest-based tree-to-string decoding problem. Our decoding algorithm consists of four steps:

1) Convert the complete parse forest to a non-complete one by introducing virtual nodes.

2) Convert the non-complete parse forest into a translation forest[2] TF by using the translation rules and the pattern-matching algorithm presented in Mi et al. (2008).

3) Prune out redundant nodes and add auxiliary hyper-edge into the translation forest for those nodes that have either no child or no father. By this step, the translation forest TF becomes a complete forest.

4) Decode the translation forest using our translation model and a dynamic search algorithm.

5 Experiment

5.1 Experiment Data and Computation Resources

We only use the official released data from CWMT 2009. Our computation resources are quite limited; we run our system on the server with following set-up: Memory 16GB, Hard Disk 500GB, CPU 2.6 GHZ. Because of this limitation and the large amount of training data, in CWMT 2009, we could only experiment quite weak parameter settings as discussed in following sections.

5.2 Training Settings

We use the official bilingual training data from all domains excluding 2005-2008 test sets to extract rules. We first encode the Chinese Character in SBC encoding, then filter the sentence pairs that are longer than 100 characters in Chinese or 50 words in English. After that we apply a latent-annotation-based (Petrov et al. 2006) parser developed by ourselves to parse the Chinese text into packed forest. The parser was trained on CTB5 and has 83.13% F-measure on all the sentences in section 271-300 of CTB5.

GIZA++ (Och and Ney, 2003) and the heuristics "grow-diag-final-and" are used to generate *m-to-n* word alignments. For parse forest pruning (Mi et al., 2008), we utilize the Margin-based pruning algorithm presented in (Huang, 2008). Different from Mi et al. (2008) that use a static pruning threshold, our threshold is sentence-depended. For each sentence, we compute the Margin between the *n*-th best and the top 1 parse tree, then use the Margin-based pruning algorithm presented in (Huang, 2008) to do pruning. By doing so, we can guarantee to use at least all the top *n* best parse trees in the forest. However, please note that even after pruning there is still exponential number of additional trees embedded in the forest because of the sharing structure of forest.

Because of computational resource limitation, we only use 10-best as pruning threshold to extract tree-to-string rules and 1-best to extract tree sequence-to-string rules. Other parameters are set as follows: maximum number of roots in a tree sequence is 3, maximum height of a translation rule is 3, maximum number of leaf nodes is 7, maximum number of node sequences on each span is 10, and maximum number of rules extracted from one node is 10000.

---

[2] The concept of translation forest is proposed in Mi et al. (2008). It is a forest that consists of only the hyper-edges induced from translation rules.

For language model training, we use SRILM Toolkits (Stolcke, 2002). However, because of computational resource limitation, we only use the target side of bilingual training data and first 4M lines of Reuter's corpus to train a 5-gram language model.

For the MER training (Och, 2003), Koehn's MER trainer (Koehn, 2007) is modified for BLEU-SBP (Papineni et al. 2002, Chiang et al., 2008).

Our evaluation metrics is case-sensitive BLEU-SBP-4.

5.3 Decoder Settings and Results

For decoding, we first parse all the sentences from the test set into packed forests and then use 300-best as threshold to prune them. After that we use the decoder to decode the packed forests into target string with the rules extracted from training data and language model mentioned in section 5.1. Then, if there are OOV in the translation result and the OOV is a two-letter Chinese word and cannot be further segmented using the Chinese segmentation tool, we simply convert them into Chinese pronunciation, otherwise it is discarded. Finally, we force the first letter of all the sentences to be in uppercase.

We tune our feature weights on the development set (CWMT-2008 test set) and do preliminary test on SSMT-2007 test set. The results are shown in Table 1.

Test Set	Without post-process	With post-process
2008	0.2480	0.2549
2007	0.2390	0.2418

Table 1. Performance Comparison

From Table 1, we could see post-processing could help improve the performance; however the bleu increase is not stable across different test set. In CWMT-2008 it is 0.69, but in SSMT-2007 it only contributes 0.28. Another point is that we are working on a less-optimal system setting due to the computational resource limitation, which is much weaker than that in (Zhang et al. 2009). As a result, the performance would be compromised a lot.

6 Conclusion

In CWMT2009, we deploy a forest-based tree sequence-to-string system to participate in Chinese-English single system track. In this paper, we give an overall description of our system, including data preprocessing, training and decoding configuration and post-processing. The preliminary experiment results on SSMT 2007 and CWMT 2008 test set are provided.

References

Chiang, David, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. 2008. *Decomposability of translation metrics for improved evaluation and efficient algorithms.* In Proc. EMNLP 2008, pages 610-619.

Huang, Liang. 2008. Forest Reranking: *Discriminative Parsing with Non-Local Features*. ACL-HLT-08. 586-594

Huang, Liang and David Chiang. 2005. *Better k-best Parsing*. IWPT-05.

Huang, Liang and David Chiang. 2007. *Forest rescoring: Faster decoding with integrated language models*. ACL-07. 144–151

Kenser, Reinhard and Hermann Ney. 1995. *Improved backing-off for M-gram language modeling*. ICASSP-95. 181-184

Klein, Dan and Christopher D. Manning. 2001. *Parsing and Hypergraphs*. IWPT-2001.

Koehn, Philipp, F. J. Och and D. Marcu. 2003. *Statistical phrase-based translation.* HLT-NAACL-03. 127-133.

Koehn, Philipp, 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.* ACL-07. 177-180. (poster)

Liu, Yang, Qun Liu and Shouxun Lin. 2006. *Tree-to-String Alignment Template for Statistical Machine Translation*. COLING-ACL-06. 609-616.

Liu, Yang, Yun Huang, Qun Liu and Shouxun Lin. 2007. *Forest-to-String Statistical Translation Rules.* ACL-07. 704-711.

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

Mi, Haitao and Liang Huang. 2008. *Forest-based Translation Rule Extraction*. EMNLP-08. 206-214.

Och, Franz J. and Hermann Ney. 2002. *Discriminative training and maximum entropy models for statistical machine translation.* ACL-02. 295-302.

Och, Franz J.. 2003. *Minimum error rate training in statistical machine translation.* ACL-03. 160-167.

Och, Franz Josef and Hermann Ney. 2003. *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics. 29(1) 19-51.

Papineni, Kishore, Salim Roukos, ToddWard and Wei-Jing Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation.* ACL-02. 311-318.

Petrov, Slav, Leon Barrett, Romain Thibaux and DanKlein. 2006. *Learning accurate, compact, and interpretable tree annotation*. COLING-ACL-06, pages 443-440.

Stolcke, Andreas. 2002. *SRILM - an extensible language modeling toolkit.* ICSLP-02. 901-904.

Zhang, Hui, Min Zhang, Haizhou Li, Aiti Aw and Chew Lim Tan. 2009. *Forest-based Tree Sequence to String Translation Model.* ACL-IJCNLP-09.

Zhang, Min, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, Sheng Li. 2008. *A Tree Sequence Alignment-based Tree-to-Tree Translation Model.* ACL-HLT-08. 559-567.

中国中文信息学会第五届全国机器翻译研讨会（CWMT2009）评测大纲. CWMT-09.