Lexicalized Syntactic Reordering Framework for Word Alignment and Machine Translation

Chung-chi Huang, Wei-teh Chen, and Jason S. Chang

Institute of Information Systems and Application, National Tsing Hua University, Hsinchu, Taiwan 300
{u901571,weitehchen,jason.jschang}@gmail.com

Abstract. We propose a lexicalized syntactic reordering framework for cross-language word aligning and translating researches. In this framework, we first flatten hierarchical source-language parse trees into syntactically-motivated linear string representations, which can easily be input to many feature-like probabilistic models. During model training, these string representations accompanied with target-language word alignment information are leveraged to learn systematic similarities and differences in languages' grammars. At runtime, syntactic constituents of source-language parse trees will be reordered according to automatically acquired lexicalized reordering rules in previous step, to closer match word orientations of the target language. Empirical results show that, as a preprocessing component, bilingual word aligning and translating tasks benefit from our reordering methodology.

Keywords: word alignment, machine translation, phrase-based decoder and syntactic reordering rule.

1 Introduction

Researchers have long believed that syntactic analyses of languages will improve natural language processing tasks, such as semantic understanding, word alignment and machine translation. In cross-lingual applications, much work has explicitly introduced grammars/models to describe/capture languages' structural divergences.

[1] is one of the pioneering researches in ordering corresponding grammatical constituents of two languages. Wu devises binary Inversion Transduction Grammar (ITG) rules to accommodate similar (in order) and different (reverse order) word orientations in synchronous bilingual parsing. The constraint imposed by Wu's straight and inverted binary branching rules is better than IBM one without syntactic insights in terms of machine translation (see [2]). On the other hand, [3], given source-language (SL) production rules of arbitrary length, utilizes EM algorithm to distinguish statistically more probable reordered grammatical sequences in target-language (TL) end from others. Recently, ever since Chiang's hierarchical phrase-based machine translation model [4], successfully integrating bilingual grammar-like rewrite rules into MT, more and more researchers have devoted themselves to syntax-based MT system: [5], [6], and [7].

W. Li and D. Mollá-Aliod (Eds.): ICCPOL 2009, LNAI 5459, pp. 103–111, 2009.

Syntactic reordering plays a vital role for modeling languages' preferences in word order in above grammatically motivated systems and has been proved to be quite effective in translation. In [8], they manually craft reordering rules concerning some characteristic differences of SL and TL word orders. These rules are aimed to reorder SL sentences such that new sequences of words better match their TL counterparts. Although better translation quality is obtained, two issues are worth mentioning: there might be exceptions to reordering rules with coarse-grained grammatical labels and their reordering rules are not automatically learnt. To address these issues, in this paper, we propose a framework which automatically acquires lexicalized reordering rules based on a parallel corpus.

The reminder of this paper is organized as follows. Section 2 discusses the reordering framework in detail. Section 3 shows the data sets used and experimental results. At last, Section 4 concludes this paper.

2 Reordering Framework

In this section, we begin with an example to illustrate how lexicalized reordering rules can have positive influence on word aligning and machine translating quality. Thereafter, we elaborate on the proposed automatic reordering framework.

2.1 An Example

Consider an English sentence "After the meeting, Mr. Chang went straight home" and its Mandarin Chinese translation "會議 後 ， 張 先生 直接 回 家". Figure 1 shows the parse tree of the English sentence (we can ignore the '· ' between English words for now), and correspondence links between the English words and their Mandarin counterparts.

In Figure 1, there are three crossings among the word alignment links, indicating three instances of reversing of some syntactic constituents during translation process. The first crossing involves the reversal of a prepositional/subordinating word and a

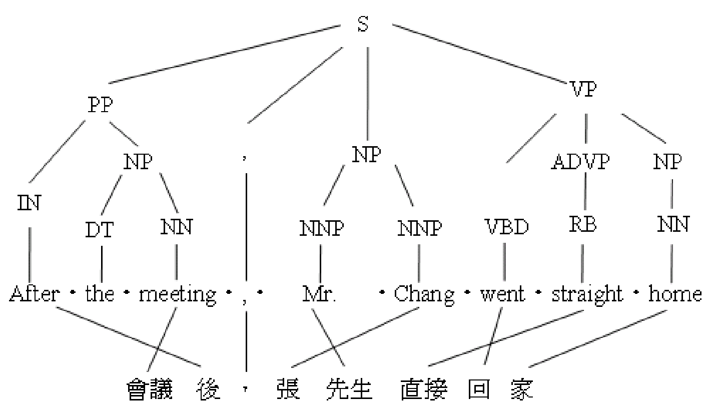

Fig. 1. An example sentence pair

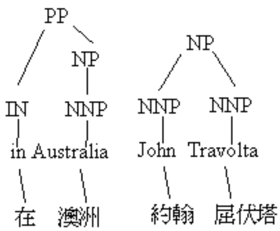

Fig. 2. Counterexamples

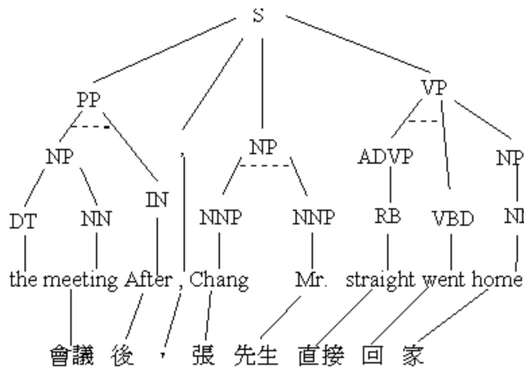

Fig. 3. Reordered tree with Chinese sentence

noun phrase. The example may lead us to conclude that we should invert all prepositional phrases when encountering IN and NP constituents. However, as indicated by the counterexample in Figure 2, the reversal of IN and NP under PP does not exist when the lexical item of IN is “in”, and syntactic information alone is not sufficient to make such a reordering decision, especially, presented with a coarse-grained grammatical label set. Instead, we further need lexical cues of the IN syntactic constituent (i.e., “after” and “in”). Accompanied with the lexicalized information, we have higher chance to recognize that, contrast to English, in Chinese, temporal subordinating conjunctions always appear after the noun phrase. Similarly, for the second crossing, we need to examine the corresponding word of the first NNP of a proper noun since a title word (e.g., “President”, “Professor”, “Mr.” and so on) has different ordering preference in Chinese, compared with the first NNP, proper noun (i.e., John), in Figure 2. [9] and [10] also point out the importance of lexical items in determining word orders from one language to another.

Encouragingly, it is straightforward to incorporate lexical information into synchronous context-free grammar rules such as ITG rules. Table 1 shows the lexicalized syntactic reordering rules that apply for the sentence pair in Figure 1. We follow Wu’s notation in [1] by using pointed bracket to depict the inverted order of the corresponding syntactic constituents in two languages and the English words enclosed in parentheses are lexical cues for the constituents. Intuitively, by learning the reordering rules shown in Table 1, we can easily transform the English tree in Figure 1 into one in Figure 3, where the horizontal dashed lines imply the subtrees had been inversely reordered. Note that such reordering rules capture languages’ divergences, thus potentially conducive to word alignment and translation.

Table 1. Lexicalized reordering rules

PP→ <IN (After) NP>
NP→ <NNP (Mr.) NNP>
VP→ <VBD ADVP>

2.2 Reordering Model

In Figure 1, we observe that the dots (·) between English words can be utilized to represent anchor points for reordering two consecutive constituents to fit the word orientations of Chinese. Take Figure 1 and Figure 3 for instance. In Figure 1, the first dot whose associated grammar rule is PP→IN NP represents the Chinese choice in ordering corresponding IN and NP syntactic constituents, containing Chinese translation of "after" and "the meeting", respectively. The fifth one whose related rule is NP→NNP NNP denotes the orientation choice of NNP and NNP counterparts in Chinese; The seventh one whose associated rule is VP→VBD ADVP represents the reordering anchor of corresponding Chinese VBD and ADVP constituents. Figure 3, on the other hand, shows the reordered tree by choosing to reverse neighboring phrases of the first, fifth and seventh anchor point in Figure 1.

Mathematically, given a SL parse tree π, reordering models search for π^*, satisfying $\arg\max_{\pi'} \Pr\left(\pi' \middle| \pi, \lambda\right)$ where λ is the set of system parameters to make syntactically reshuffled π^* more in tune with grammar in target language. In this paper, by using tree-to-string transformation algorithm illustrated below, we first transform the parse tree π into a string representation *s* with syntactic information encoded in the artificial anchor points denoted by '· '. Then, the problem of searching for most probable transformed tree (π^*) can be recast as one of finding best (linear) reordering label sequence for anchor points in *s*. In other words, provided with the representation *s* and the system parameter set $\lambda = \{\lambda_j\}$, our model looks for the most likely reordering label sequence y^*

$$y^* = \arg\max_y p\left(y \middle| s, \lambda\right) = \arg\max_y \frac{1}{z\left(s\right)} \exp\left(\sum_j \lambda_j F_j\left(y, s\right)\right). \tag{1}$$

where z(*s*) is a normalization factor and λ_j is the weight of the syntactic feature function F_j. Equation (1) is in line with the conditional nature of conditional random fields. Therefore, we feed the SL string into the probabilistic sequential labeler of conditional random fields (CRFs) to find the best label sequence, upon which most probable reordered tree are based.

Tree-to-String Transformation Algorithm

Input: source-language sentence *e* and its parse tree π
Output: string representation *s*

```
INSERT '· ' between words in e   // as reordering anchor points
FOR each word w IN e
   IF w is '· '
     Lw=the word immediate to the left of the · in e
     Rw=the word immediate to the right of the · in e
     Along π, find the closest common ancestor node P for words Lw and Rw
     LHS=the immediate descendent of node P along the path from P to Lw in π
     RHS=the immediate descendent of node P along the path from P to Rw in π
       ASSOCIATE the grammatical rule P→LHS RHS, Lw, and Rw WITH this dot
          and ROCORD this information IN s
OUTPUT s
```

Table 2. String representation of tree in Figure 1

Dot	P	LHS	Lw	RHS	Rw
$\cdot_1$	PP	IN	After	NP	the
$\cdot_2$	NP	DT	the	NN	meeting
$\cdot_3$	S	PP	meeting	,	,
$\cdot_4$	S	,	,	NP	Mr.
$\cdot_5$	NP	NNP	Mr.	NNP	Chang
$\cdot_6$	S	NP	Chang	VP	went
$\cdot_7$	VP	VBD	went	ADVP	straight
$\cdot_8$	VP	ADVP	straight	NP	home

Table 2 summarizes the content derived from abovementioned transformation algorithm on the parse tree of Figure 1 and this information will be fed to CRFs to determine the reordering tag (in order or inversion) of the anchor points. In Table 2, **dot** column stands for artificial anchor points in SL sentence, **Lw** and **Rw** for previous word and successive word of the current one respectively, and **P**, **LHS**, **Lw**, **RHS** and **Rw** constitute the syntactic reordering features of our model. Notice that, inspired by [1] and [11], we assume SL parse trees are binarized before fed into the tree-to-string transformation algorithm. [1] suggests binary-branching ITG rules prune seemingly unlikely and arbitrary word permutations but yet, at the same time, accommodate most meaningful structural reversals during translation. In [11] binarization process is reported to be beneficial to machine translation in terms of quality and speed. Therefore, in this paper we focus on reorderings of binary trees which can be obtained by binary syntactic parsers (e.g., Berkeley Parser) or by following the binarizing process in [11].

If, during training, probabilistic CRFs always observe inversion of the grammar rule PP→IN NP especially with lexical "after" presenting in Lw field, and straight order of the rule S→NP VP, CRFs model will tag the first dot as I (for inversion) and tag the sixth as S (for straightness). Moreover, if the reordering tag of every dot is correctly determined, the SL parse tree in Figure 1 will be successfully reordered into one in Figure 3, which abides by grammatical ordering preferences in the target language (e.g., Chinese).

Our framework leverages CRFs to train the weights of the feature functions related to syntactic labels, syntactic rules, and lexical items provided by our tree-to-string transformation procedure, and, at runtime, brings SL parse trees closer to TL word order by applying lexicalized reordering grammar rules or pure grammatical rules. Which type to choose is informed by highly-tuned feature weights in CRFs training.

2.3 Training Process of CRFs

To gain insights on how to order the corresponding SL syntactic constituents in the target language, SL sentences are aligned to TL sentences at word level and are monolingually parsed by some existing parser. Furthermore, based on word alignment results, firstly, the minimum/maximum word position on target language end that SL part-of-speech tags can cover is determined, i.e. TL spans of SL POS tags, and then the TL spans are iteratively obtained in bottom-up fashion. In the end, parse trees contain not only monolingual grammatical labels but also bilingual information concerning the TL span of each SL tree node, on which we work to differentiate dissimilar word ordering preferences in the two languages from similar ones. The training process is outlined as below.

Training Procedure

Input: a sentence-aligned corpus **C**={(*e*,*f*)}, a word aligner *WA*, a source language parser *Par*, and a CRFs implementation *crf*
Output: system parameters of our reordering model λ
APPLY *WA* on the corpus **C** to obtain word alignment
PARSE source-language end of **C** by use of *Par*
FOR each sentence pair (*e*,*f*) IN **C**
  DENOTE π as the parse tree of *e*, π (*pos*) as the part-of-speech nodes in π and π (*nonT*) as the syntactic (non-terminal) nodes in π excluding nodes in π (*pos*)
  FOR each node *n* IN π (*pos*)
  span(*n*)=from min(aligned positions on TL side of *n*) to max(aligned positions on TL side of *n*)
  FOR each node *n* IN π (*nonT*)
  span(*n*)=from min(aligned positions on TL side of *n*'s children) to max(aligned positions on TL side of *n*'s children)
  APPLY tree-to-string transformation algorithm on *e* and π to obtain their string representation *s*
  FOR each dot *d* IN *s*
    IF span(*d* 's LHS) <[1] span(*d* 's RHS)
     APPEND the orientation tag 'S' to *d* // straight order
    IF span(*d* 's LHS) >[2] span(*d* 's RHS)
     APPEND the orientation tag 'I' to *d* // inverted order

After collecting all the string representations with ordering information, we train *crf* to determine the weights λ associated with chosen syntactic feature functions.

Take sentence pair in Figure 1 for example. The TL-end span of each label in the parse tree and the string representation with orientation information are shown in Figure 4 and in Table 3 respectively. String representations with orientation information of sentences are leverage to tune the system weights $\lambda=\{\lambda_j\}$. The weights reflect the contribution of lexical or syntactic items in determining TL word orders of a specific SL context (e.g., PP→ <IN (After) NP (the)>).

Table 3. String representation with ordering tags

Dot	P	LHS	RHS	Order
$\cdot_1$	PP	IN	NP	I
$\cdot_2$	NP	DT	NN	S[3]
$\cdot_3$	S	PP	,	S
$\cdot_4$	S	,	NP	S
$\cdot_5$	NP	NNP	NNP	I
$\cdot_6$	S	NP	VP	S
$\cdot_7$	VP	VBD	ADVP	I
$\cdot_8$	VP	ADVP	NP	S

[1] The min index of span of the second operand is larger than the max index of the first one.
[2] The min index of span of the first operand is larger than the max index of the second one.
[3] Intuitively, if one of the spans is NULL, straight order is adopted.

span(IN)=from 2 to 2
span(DT)=NULL
span(NN)=from 1 to 1
span(,)=from 3 to 3
span(NNP-Mr.)=from 5 to 5
span(NNP-Chang)=from 4 to 4
span(VBD)=from 7 to 7
span(RB)=from 6 to 6
span(NN)=from 8 to 8
span(NP-the meeting)=from 1 to 1
span(NP-Mr. Chang)=from 4 to 5
span(ADVP)=from 6 to 6
span(NP-home)=from 8 to 8
span(PP)=from 1 to 2
span(VP)=from 6 to 8
span(S)=from 1 to 8

Fig. 4. TL-end span of each label

3 Experiments

We start with the data sets and settings we used in experiments. Afterwards, we evaluate the impact our reordering framework has on performance of bilingual word alignment and machine translation.

3.1 Data Sets and Experimental Settings

We used the first 200,000 sentence pairs of the news portion of Hong Kong Parallel Text as our parallel corpus **C**. A MT testing data set, composed of 1035 English sentences of average 28 words randomly chosen from Hong Kong news[4] (excluding sentences in **C**), was allocated. The corresponding Chinese sentences made up of its reference translation set, that is, one reference translation per English sentence. Moreover, the English sentences in both training and testing sets were syntactically parsed by Berkeley parser[5] beforehand.

We employed CRF++[6] as the implementation of probabilistic conditional random fields to construct the proposed syntactic reordering framework. During CRFs' parameters training (Section 2.3), we deployed GIZA++ as the word aligner. Besides, to make CRF++ more accurately learn ordering choices of two languages in syntactic constituents, sentence pairs in **C** would *not* be utilized if the word alignment rate of content words (nouns, verbs and adjectives) on English end was lower than 0.8 or the

[4] The news portion of Hong Kong Parallel Text.
[5] http://nlp.cs.berkeley.edu/Main.html
[6] It is freely distributed in http://crfpp.sourceforge.net/

length of the English sentence was shorter than 20. In other words, CRF++ was dedicated to search for significant lexicalized or non-lexicalized reordering rules from highly-aligned and potentially long-range distorted sentence pairs. After filtering, approximately 23,000 parallel sentences of **C** were retained to tune CRF++.

At runtime translation, on the other hand, our framework exploited Pharaoh ([12]) as the phrase-based MT decoder. The language model Pharaoh needs was trained on the Chinese part of the whole Hong Kong news, 739,919 sentences in total, using SRI language modeling toolkit, while phrase translation table was built upon **C** after word aligned using GIZA++.

3.2 Evaluation

We are interested in examining whether our methodology captures meaningful syntactic relationships between the source and target languages, thus boosting the accuracy in word alignment and decoding. We experimented different ways of introducing source sentence reordering to the phrase-based machine translation system (i.e., Pharaoh). First, we performed word alignment on the original and reordered source sentences to derive two sorts of phrase translation table used in MT decoder. Then decoder was run on the unaltered test sentences as well as reordered test sentences. Therefore, there are four sets of translation results where the source sentences in the training data and test data are either unaltered or reordered. The translation quality using these four data sets was measured by BLEU scores ([13]) and summarized in a contingency matrix in Table 4.

Table 4. Results of translation quality

	original training data	reordered training data
original testing data	23.43	24.16
reordered testing data	24.76	25.71

As suggested by Table 4, when using the reordered sentences to perform word alignment and decoding, the translation quality improved by more than 0.7 BLEU point. If we left the training data unchanged and simply reordered the test sentences, we get a significant improvement of 1.3 BLEU points over translating the original test sentences. One can find that test sentence reordering resulted in greater improvement (6% relative) over training sentence reordering (3% relative). There might be two reasons for this difference. Firstly, our result is consistent with the observation presented by [14]: it is, sometimes, difficult to propagate improvements in word alignment to translation. Additionally, GIZA++, a word aligner modeling distortion in languages, is much more capable of capturing distortion of words than Pharaoh, a decoder exhibiting global reordering problems. As a result, there were about 3% improvement gap between these two different settings of data sets.

Encouragingly, if both the training and test sentences were pre-reordered, our method outperformed baseline by more than 2 BLEU points. Overall, it is safe to say that our automated reordering framework improves translation quality for disparate language pair such as English and Chinese.

4 Conclusion and Future Work

This paper has introduced a syntactic reordering framework which automatically learns reordering rules from a parallel corpus using conditional random fields. In experiments, these reordering rules, if necessary, accompanied with lexical information, are proved to be conducive to relieving the pressure of distortion modeling off word aligners and MT systems alike.

As for future work, we would like to examine whether integrating more syntactic features (e.g. the height of a tree node, the head of the phrase and etc.) into the framework further boosts the performance. We also like to inspect the performance of our methodology in other distantly-related language pairs such as English and Arabic.

References

1. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics 23(3), 377–403 (1997)
2. Zens, R., Ney, H.: A comparative study on reordering constraints in statistical machine translation. In: ACL, pp. 144–151 (2003)
3. Yamada, K., Knight, K.: A syntax-based statistical translation model. In: ACL, pp. 523–530 (2001)
4. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: ACL, pp. 263–270 (2005)
5. Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., Thayer, I.: Scalable Inference and Training of Context-Rich Syntactic Translation Models. In: ACL, pp. 961–968 (2006)
6. Zhang, D., Li, M., Li, C.-h., Zhou, M.: Phrase reordering model integrating syntactic knowledge for SMT. In: EMNLP/CoNLL, pp. 533–540 (2007)
7. Liu, Y., Huang, Y., Liu, Q., Lin, S.: Forest-to-string statistical translation rules. In: ACL, pp. 704–711 (2007)
8. Wang, C., Collins, M., Koehn, P.: Chinese syntactic reordering for statistical machine translation. In: EMNLP/CoNLL, pp. 737–745 (2007)
9. Xiong, D., Liu, Q., Lin, S.: Maximum entropy based phrase reordering model for statistical machine translation. In: ACL/COLING, pp. 521–528 (2006)
10. Zhang, H., Gildea, D.: Stochastic lexicalized inversion transduction grammar for alignment. In: ACL, pp. 475–482 (2005)
11. Zhang, H., Huang, L., Gildea, D., Knight, K.: Synchronous binarization for machine translation. In: NAACL/HLT, pp. 256–263 (2006)
12. Koehn, P., Och, F., Marcu, D.: Statistical phrase-based translation. In: NAACL/HLT, pp. 48–54 (2003)
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.-j.: BLEU: a method for automatic evaluation of machine translation. In: ACL, pp. 311–318 (2002)
14. Ayan, N.F., Dorr, B.J.: Going beyond AER: an extensive analysis of word alignments and their impact on MT. In: ACL, pp. 9–16 (2006)