A non-contiguous Tree Sequence Alignment-based Model for Statistical Machine Translation

Jun Sun[1,2] **Min Zhang**[1] **Chew Lim Tan**[2]
[1] Institute for Infocomm Research [2]School of Computing, National University of Singapore
sunjun@comp.nus.edu.sg mzhang@i2r.a-star.edu.sg tancl@comp.nus.edu.sg

Abstract

The tree sequence based translation model allows the violation of syntactic boundaries in a rule to capture non-syntactic phrases, where a tree sequence is a contiguous sequence of sub-trees. This paper goes further to present a translation model based on non-contiguous tree sequence alignment, where a non-contiguous tree sequence is a sequence of sub-trees and gaps. Compared with the contiguous tree sequence-based model, the proposed model can well handle non-contiguous phrases with any large gaps by means of non-contiguous tree sequence alignment. An algorithm targeting the non-contiguous constituent decoding is also proposed. Experimental results on the NIST MT-05 Chinese-English translation task show that the proposed model statistically significantly outperforms the baseline systems.

1 Introduction

Current research in statistical machine translation (SMT) mostly settles itself in the domain of either phrase-based or syntax-based. Between them, the phrase-based approach (Marcu and Wong, 2002; Koehn et al, 2003; Och and Ney, 2004) allows local reordering and contiguous phrase translation. However, it is hard for phrase-based models to learn global reorderings and to deal with non-contiguous phrases. To address this issue, many syntax-based approaches (Yamada and Knight, 2001; Eisner, 2003; Gildea, 2003; Ding and Palmer, 2005; Quirk et al, 2005; Zhang et al, 2007, 2008a; Bod, 2007; Liu et al, 2006, 2007; Hearne and Way, 2003) tend to integrate more syntactic information to enhance the non-contiguous phrase modeling. In general, most of them achieve this goal by introducing syntactic non-terminals as translational equivalent placeholders in both source and target sides. Nevertheless, the generated rules are strictly required to be derived from the *contiguous* translational equivalences (Galley et al, 2006; Marcu et al, 2006; Zhang et al, 2007, 2008a, 2008b; Liu et al, 2006, 2007). Among them, Zhang et al. (2008a) acquire the non-contiguous phrasal rules from the *contiguous* tree sequence pairs[1], and find them useless via real syntax-based translation systems. However, Wellington et al. (2006) statistically report that discontinuities are very useful for translational equivalence analysis using binary branching structures under word alignment and parse tree constraints. Bod (2007) also finds that discontinues phrasal rules make significant improvement in linguistically motivated STSG-based translation model. The above observations are conflicting to each other. In our opinion, the non-contiguous phrasal rules themselves may not play a trivial role, as reported in Zhang et al. (2008a). We believe that the effectiveness of non-contiguous phrasal rules highly depends on how to extract and utilize them.

To verify the above assumption, suppose there is only one tree pair in the training data with its alignment information illustrated as Fig. 1(a)[2]. A test sentence is given in Fig. 1(b): the source sentence with its syntactic tree structure as the upper tree and the expected target output with its syntactic structure as the lower tree. In the tree sequence alignment based model, in addition to the entire tree pair, it is capable to acquire the contiguous tree sequence pairs: TSP (1~4)[3] in Fig. 1. By means of the rules derived from these contiguous tree sequence pairs, it is easy to translate the contiguous phrase "他/he 出场/show up 的/'s". As for the non-contiguous phrase "到/at, ***, 时候/time", the only related rule is r_1 derived from TSP4 and the entire tree pair. However, the source side of r_1 does not match the source tree structure of the test sentence. Therefore, we can only partially translate the illustrated test sentence with this training sample.

[1] A tree sequence pair in this context is a kind of translational equivalence comprised of a pair of tree sequences.
[2] We illustrate the rule extraction with an example from the tree-to-tree translation model based on tree sequence alignment (Zhang et al, 2008a) without losing of generality to most syntactic tree based models.
[3] We only list the contiguous tree sequence pairs with one single sub-tree in both sides without losing of generality.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 914–922,
Suntec, Singapore, 2-7 August 2009.

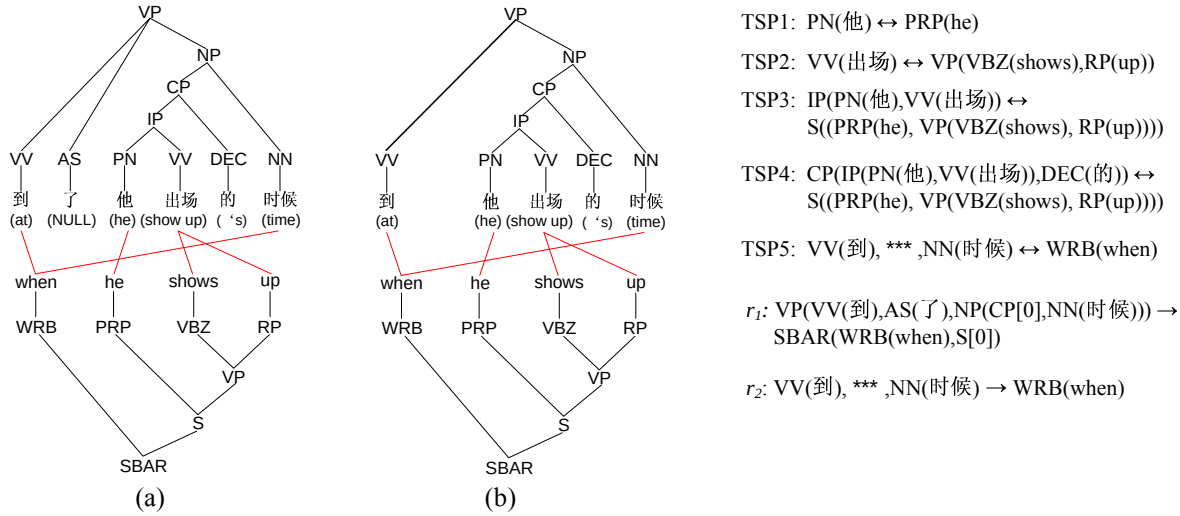

Figure 1: Rule extraction of tree-to-tree model based on tree sequence pairs

As discussed above, the problem lies in that the non-contiguous phrases derived from the contiguous tree sequence pairs demand greater reliance on the context. Consequently, when applying those rules to unseen data, it may suffer from the data sparseness problem. The expressiveness of the model also slacks due to their weak ability of generalization.

To address this issue, we propose a syntactic translation model based on non-contiguous tree sequence alignment. This model extracts the translation rules not only from the *contiguous* tree sequence pairs but also from the *non-contiguous* tree sequence pairs where a non-contiguous tree sequence is a sequence of sub-trees and gaps. With the help of the non-contiguous tree sequence, the proposed model can well capture the *non-contiguous* phrases in avoidance of the constraints of large applicability of context and enhance the non-contiguous constituent modeling. As for the above example, the proposed model enables the non-contiguous tree sequence pair indexed as TSP5 in Fig. 1 and is allowed to further derive r_2 from TSP5. By means of r_2 and the same processing to the contiguous phrase "他/he 出场/show up 的/'s" as the contiguous tree sequence based model, we can successfully translate the entire source sentence in Fig. 1(b).

We define a synchronous grammar, named Synchronous non-contiguous Tree Sequence Substitution Grammar (SncTSSG), extended from synchronous tree substitution grammar (STSG: Chiang, 2006) to illustrate our model. The proposed synchronous grammar is able to cover the previous proposed grammar based on tree (STSG, Eisner, 2003; Zhang et al, 2007) and tree sequence (STSSG, Zhang et al, 2008a) alignment. Besides, we modify the traditional parsing based decoding algorithm for syntax-based SMT to facilitate the non-contiguous constituent decoding for our model.

To the best of our knowledge, this is the first attempt to acquire the translation rules with rich syntactic structures from the non-contiguous Translational Equivalences (non-contiguous tree sequence pairs in this context).

The rest of this paper is organized as follows: Section 2 presents a formal definition of our model with detailed parameterization. Sections 3 and 4 elaborate the extraction of the non-contiguous tree sequence pairs and the decoding algorithm respectively. The experiments we conduct to assess the effectiveness of the proposed method are reported in Section 5. We finally conclude this work in Section 6.

2 Non-Contiguous Tree sequence Alignment-based Model

In this section, we give a formal definition of SncTSSG and accordingly we propose the alignment based translation model. The details of probabilistic parameterization are elaborated based on the log-linear framework.

2.1 Synchronous non-contiguous TSSG (SncTSSG)

Extended from STSG (Shiever, 2004), SncTSSG can be formalized as a quintuple G = <Σ_s , Σ_t, N_s, N_t, R>, where:

- Σ_s and Σ_t are source and target terminal alphabets (words) respectively, and
- N_s and N_t are source and target non-terminal alphabets (linguistically syntactic tags, i.e. NP, VP) respectively; as well as the non-terminal $< *** >$ to denote a gap,

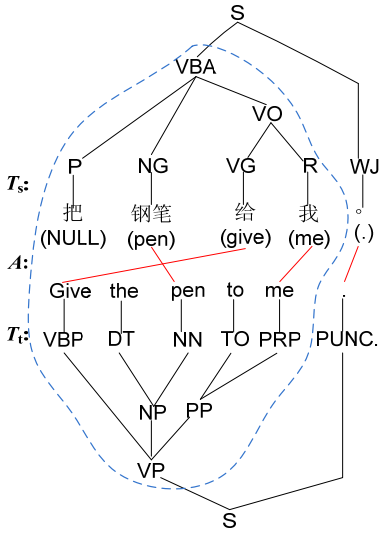

Figure 2: A word-aligned parse tree pair

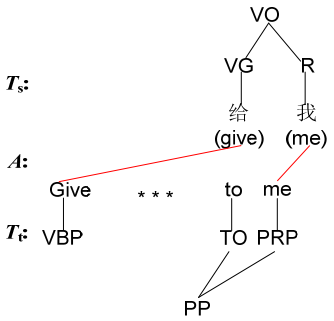

Figure 3: A *non-contiguous* tree sequence pair

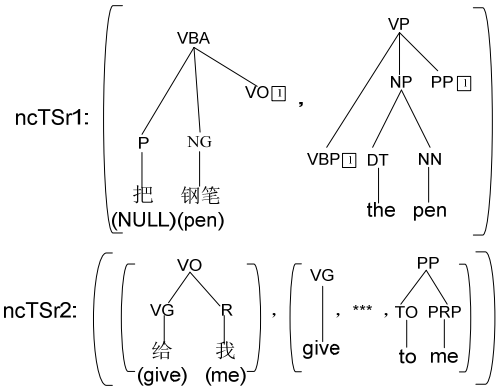

Figure 4: Two examples of *non-contiguous* tree sequence translation rules

$< *** >$ can represent any syntactic or non-syntactic tree sequences, and

- R is a production rule set consisting of rules derived from corresponding contiguous or non-contiguous tree sequence pairs, where a rule is a pair of contiguous or non-contiguous tree sequence with alignment relation between leaf nodes across the tree sequence pair.

A non-contiguous tree sequence translation rule $r \in \mathrm{R}$ can be further defined as a triple $r =< TS(f(M_{j_1^k}^{j_2^k})), TS(e(N_{i_1^l}^{i_2^l})), \tilde{A} >$, where:

- $TS(f(M_{j_1^k}^{j_2^k}))$is a non-contiguous source tree sequence, covering the span set $M = \{[j_1^k, j_2^k] | k = 1, \ldots, m\}$ in $T(f_1^J)$, where $j_1^t < j_2^t$ which means each subspan has non-zero width and $j_2^t < j_1^{t+1}$ which means there is a non-zero gap between each pair of consecutive intervals. A gap of interval $[j_2^t, j_1^{t+1}]$ is denoted as $< *** >$, and
- $TS(e(N_{i_1^l}^{i_2^l}))$is a non-contiguous target tree sequence, covering the span set $N = \{[i_1^l, i_2^l] | l = 1, \ldots, n\}$ in $T(e_1^I)$, where $i_1^t < i_2^t$ which means each subspan has non-zero width and $i_2^t < i_1^{t+1}$ which means there is a non-zero gap between each pair of consecutive intervals. A gap of interval $[i_2^t, i_1^{t+1}]$ is denoted as $< *** >$, and
- $\tilde{A}$ are the alignments between leaf nodes of the source and target non-contiguous tree sequences, satisfying the following conditions :

$$\forall (i, j) \in \tilde{A}: i_1^l \le i \le i_2^l \leftrightarrow j_1^k \le j \le j_2^k,$$

where $[i_1^l, i_2^l] \in N$ and $[j_1^k, j_2^k] \in M$

In SncTSSG, the leaf nodes in a non-contiguous tree sequence rule can be either non-terminal symbols (grammar tags) or terminal symbols (lexical words) and the non-terminal symbols with the same index which are subsumed simultaneously are not required to be contiguous. Fig. 4 shows two examples of non-contiguous tree sequence rules ("non-contiguous rule" for short in the following context) derived from the non-contiguous tree sequence pair (in Fig. 3) which is extracted from the bilingual tree pair in Fig. 2. Between them, ncTSr1 is a tree rule with internal nodes non-contiguously subsumed from a contiguous tree sequence pair (dashed in Fig. 2) while ncTSr2 is a non-contiguous rule with a contiguous source side and a non-contiguous target side. Obviously, the non-contiguous tree sequence rule ncTSr2 is more flexible by neglecting the context among the gaps of the tree sequence pair while capturing all aligned counterparts with the corresponding syntactic structure information. We

expect these properties can well address the issues of non-contiguous phrase modeling.

2.2 SncTSSG based Translation Model

Given the source and target sentence f_1^J and e_1^I, as well as the corresponding parse trees $T(f_1^J)$ and $T(e_1^I)$, our approach directly approximates the posterior probability $Pr(T(e_1^I)|T(f_1^J))$ based on the log-linear framework:

$$\boldsymbol{Pr(T(e_1^I)|T(f_1^J)) =}$$

$$\frac{1}{Z_{T(f_1^J)}} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(TS(e(N_{i_p}^{i_q})), TS(f(M_{j_k}^{j_l})), \tilde{A})\right)$$

In this model, the feature function h_m is log-linearly combined by the corresponding parameter λ_m(Och and Ney, 2002). The following features are utilized in our model:

1) The bi-phrasal translation probabilities
2) The bi-lexical translation probabilities
3) The target language model
4) The # of words in the target sentence
5) The # of rules utilized
6) The average tree depth in the source side of the rules adopted
7) The # of non-contiguous rules utilized
8) The # of reordering times caused by the utilization of the non-contiguous rules

Feature 1~6 can be applied to either STSSG or SncTSSG based models, while the last two targets SncTSSG only.

3 Tree Sequence Pair Extraction

In training, other than the contiguous tree sequence pairs, we extract the non-contiguous ones as well. Nevertheless, compared with the contiguous tree sequence pairs, the non-contiguous ones suffer more from the *tree sequence pair redundancy* problem that one non-contiguous tree sequence pair can be comprised of two or more unrelated and nonadjacent contiguous ones. To model the contiguous phrases, this problem is actually trivial, since the contiguous phrases stay adjacently and share the related syntactic constraints; however, as for non-contiguous phrase modeling, the cohesion of syntactically and semantically unrelated tree sequence pairs is more likely to generate noisy rules which do not benefit at all. In order to minimize the number of redundant tree sequence pairs, we limit the # of gaps of non-contiguous tree sequence pairs to be 0 in either source or target side. In other words, we only allow one side to be non-contiguous (either source or target side) to partially reserve its syntactic and semantic cohesion[4]. We further design a two-phase algorithm to extract the tree sequence pairs as described in Algorithm 1.

Algorithm 1: *Tree Sequence Pair Extraction*

Input: *source tree and target tree*
Output: *the set of tree sequence pairs*

Data structure:
$p[j_1, j_2]$ to store tree sequence pairs covering source span$[j_1, j_2]$

1: **foreach** source span $[j_1, j_2]$, **do**
2: find a target span $[i_1,i_2]$ with minimal length covering all the target words aligned to $[j_1, j_2]$
3: **if** all the target words in $[i_1,i_2]$ are aligned with source words *only* in $[j_1, j_2]$, **then**
4: Pair each source tree sequence covering $[j_1, j_2]$ with those in target covering $[i_1,i_2]$ as a contiguous tree sequence pair
5: Insert them into $p[j_1, j_2]$
6: **else**
7: create sub-span set s($[i_1,i_2]$) to cover all the target words aligned to $[j_1, j_2]$
8: Pair each source tree sequence covering $[j_1, j_2]$ with each target tree sequence covering s($[i_1,i_2]$) as a non-contiguous tree sequence pair
9: Insert them into $p[j_1, j_2]$
10: **end if**
11:**end do**
12: **foreach** target span $[i_1,i_2]$, **do**
13: find a source span $[j_1, j_2]$ with minimal length covering all the source words aligned to $[i_1,i_2]$
14: **if** any source word in $[j_1, j_2]$ is aligned with target words outside $[i_1,i_2]$, **then**
15: create sub-span set s($[j_1, j_2]$) to cover all the source words aligned to $[i_1,i_2]$
16: Pair each source tree sequence covering s($[j_1, j_2]$) with each target tree sequence covering $[i_1,i_2]$ as a non-contiguous tree sequence pair
17: Insert them into $p[j_1, j_2]$
18: **end if**
19: **end do**

For the first phase (line 1-11), we extract the contiguous tree sequence pairs (line 3-5) and the non-contiguous ones with contiguous tree sequence in the source side (line 6-9). In the second phase (line 12-19), the ones with contiguous tree sequence in the target side and non-contiguous tree sequence on the source side are extracted.

[4] Wellington et al. (2006) also reports that allowing gaps in one side only is enough to eliminate the hierarchical alignment failure with word alignment and *one side* parse tree constraints. This is a particular case of our definition of non-contiguous tree sequence pair since a non-contiguous tree sequence can be considered to overcome the structural constraint by neglecting the structural information in the gaps.

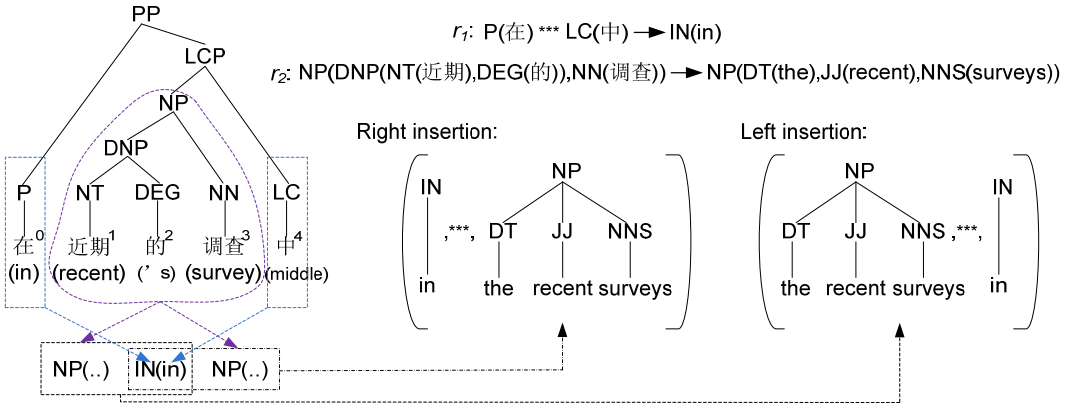

Figure 5: Illustration of "Source gap insertion"

The extracted tree sequence pairs are then utilized to derive the translation rules. In fact, both the contiguous and non-contiguous tree sequence pairs themselves are applicable translation rules; we denote these rules as *Initial* rules. By means of the *Initial* rules, we derive the *Abstract* rules similarly as in Zhang et al. (2008a).

Additionally, we develop a few constraints to limit the number of *Abstract* rules. The depth of a tree in a rule is no greater than h. The number of non-terminals as leaf nodes is no greater than c. The tree number is no greater than d. Besides, the number of lexical words at leaf nodes in an *Initial* rule is no greater than l. The maximal number of gaps for a non-contiguous rule is no greater than g.

4 The Pisces decoder

We implement our decoder Pisces by simulating the span based CYK parser constrained by the rules of SncTSSG. The decoder translates each span iteratively in a bottom up manner which guarantees that when translating a source span, any of its sub-spans is already translated.

For each source span $[j_1, j_2]$, we perform a three-phase decoding process. In the first phase, the source side contiguous translation rules are utilized as described in Algorithm 2. When translating using a source side contiguous rule, the target tree sequence of the rule whether contiguous or non-contiguous is directly considered as a candidate translation for this span (line 3), if the rule is an *Initial* rule; otherwise, the non-terminal leaf nodes are replaced with the corresponding sub-spans' translations (line 5).

In the second phase, the source side non-contiguous rules[5] for $[j_1, j_2]$ are processed. As for the ones with non-terminal leaf nodes, the replacement with corresponding spans' translations is initially performed in the same way as with the contiguous rules in the first phase. After that, an operation specified for the source side non-contiguous rules named "Source gap insertion" is performed. As illustrated in Fig. 5, to use the non-contiguous rule r_1, which covers the source span set ([0,0], [4,4]), the target portion "IN(in)" is first attained, then the translations to the gap span [1,3] is acquired from the previous steps and is inserted either to the right or to the left of "IN(in)". The insertion is rather cohesion based but leaves a *gap* <***> for further "Target tree sequence reordering" in the next phase if necessary.

Algorithm 2: *Contiguous rule processing*

Data structure:
$h[j_1, j_2]$to store translations covering source span$[j_1, j_2]$

1: **foreach** rule r contiguous in source span $[j_1, j_2]$, **do**
2: **if** r is an *Initial* rule, **then**
3: insert r into $h[j_1, j_2]$
4: **else** //*Abstract* rule
5: generate translations by replacing the non-terminal leaf nodes of r with their corresponding spans' translation
6: insert the new translation into $h[j_1, j_2]$
7: **end if**
8: **end do**

In the third phase, we carry out the other non-contiguous rule specific operation named "Target tree sequence reordering". Algorithm 3 gives an overview of this operation. For each source span, we first binarize the span into the left one and the right one. The translation hypothesis for this span is generated by firstly inserting the candidate translations of the right span to each gap in the ones of the left span respectively (line 2-9) and then repeating in the alternative direction (line10-17). The gaps for the insertion of the tree sequences in the target side are generated from either the inherit-

[5] A source side non-contiguous translation rules which cover a list of n non-contiguous spans s($[k_1^i, k_2^i]$, i=1,…,n) is considered to cover the source span $[j_1, j_2]$ if and only if $k_1^1 = j_1$ and $k_2^n = j_2$.

Algorithm 3: *Target tree sequence reordering*

Data structure:
$h[j_1, j_2]$to store translations covering source span$[j_1, j_2]$

```
1: foreach k ∈[j1, j2), do
2:    foreach translation t̂l ∈ h[j1, k], do
3:       foreach gap ĝl in t̂l, do
4:          foreach translation t̂r ∈ h[k+1, j2], do
5:             insert t̂r into the position of ĝl
6:             insert the new translation into h[j1, j2]
7:          end do
8:       end do
9:    end do
10:   foreach translation t̂r ∈ h[k+1, j2], do
11:      foreach gap ĝr in t̂r, do
12:         foreach translation t̂l ∈ h[j1, k], do
13:            insert t̂l into the position of ĝr
14:            insert the new translation into h[j1, j2]
15:         end do
16:      end do
17:   end do
18:end do
```

ance of the target side non-contiguous tree sequence pairs or the production of the previous operations of "Source gap insertion". Therefore, the insertion for target gaps helps search for a better order of the non-contiguous constituents in the target side. On the other hand, the non-contiguous tree sequences with rich syntactic information are reordered, nevertheless, without much consideration of the constraints of the syntactic structure. Consequently, this distortional operation, like phrase-based models, is much more flexible in the order of the target constituents than the traditional syntax-based models which are limited by the syntactic structure. As a result, "Target tree sequence reordering" enhances the reordering ability of the model.

To speed up the decoder, we use several thresholds to limit the searching space for each span. The maximal number of the rules in a source span is no greater than α. The maximal number of translation candidates for a source span is no greater than β. On the other hand, to simplify the computation of language model, we only compute for source side contiguous translational hypothesis, while neglecting gaps in the target side if any.

5 Experiments

5.1 Experimental Settings

In the experiments, we train the translation model on FBIS corpus (7.2M (Chinese) + 9.2M (English) words) and train a 4-gram language model on the Xinhua portion of the English Gigaword corpus (181M words) using the SRILM Toolkits (Stolcke, 2002). We use these sentences with less than 50 characters from the NIST MT-2002 test set as the development set and the NIST MT-2005 test set as our test set. We use the Stanford parser (Klein and Manning, 2003) to parse bilingual sentences on the training set and Chinese sentences on the development and test set. The evaluation metric is case-sensitive BLEU-4 (Papineni et al., 2002). We base on the *m*-to-*n* word alignments dumped by GIZA++ to extract the tree sequence pairs. For the MER training, we modify Koehn's version (Koehn, 2004). We use Zhang et al's implementation (Zhang et al, 2004) for 95% confidence intervals significant test.

We compare the SncTSSG based model against two baseline models: the phrase-based and the STSSG-based models. For the phrase-based model, we use Moses (Koehn et al, 2007) with its default settings; for the STSSG and SncTSSG based models we use our decoder Pisces by setting the following parameters: $d = 4$, $h = 6$, $c = 6$, $l = 6$, $\alpha = 50$, $\beta = 50$. Additionally, for STSSG we set $g = 0$, and for SncTSSG, we set $g = 1$.

5.2 Experimental Results

Table 1 compares the performance of different models across the two systems. The proposed SncTSSG based model significantly outperforms ($p < 0.05$) the two baseline models. Since the SncTSSG based model covers the STSSG based model in its modeling ability and obtains a superset in rules, the improvement empirically verifies the effectiveness of the additional non-contiguous rules.

System	Model	BLEU
Moses	cBP	23.86
Pisces	STSSG	25.92
	SncTSSG	26.53

Table 1: Translation results of different models (**cBP** refers to contiguous bilingual phrases without syntactic structural information, as used in Moses)

Table 2 measures the contribution of different combination of rules. **cR** refers to the rules derived from contiguous tree sequence pairs (i.e., all STSSG rules); **ncPR** refers to non-contiguous phrasal rules derived from *contiguous* tree sequence pairs with at least one non-terminal leaf node between two lexicalized leaf nodes (i.e., all non-contiguous rules in STSSG defined as in Zhang et al. (2008a)); **srcncR** refers to source side non-contiguous rules (SncTSSG rules only, not STSSG rules); **tgtncR** refers to target side non-contiguous rules (SncTSSG rules only, not STSSG rules) and **src&tgtncR** refers non-contiguous rules

ID	Rule Set	BLEU
1	cR (STSSG)	25.92
2	cR w/o ncPR	25.87
3	cR w/o ncPR + tgtncR	26.14
4	cR w/o ncPR + srcncR	26.50
5	cR w/o ncPR + src&tgtncR	26.51
6	cR + tgtncR	26.11
7	cR + srcncR	26.56
8	cR+src&tgtncR(SncTSSG)	26.53

Table 2: Performance of different rule combination

with gaps in either side (**srcncR+ tgtncR**). The last three kinds of rules are all derived from *non-contiguous* tree sequence pairs.

1) From Exp 1 and 2 in Table 2, we find that non-contiguous phrasal rules (**ncPR**) derived from contiguous tree sequence pairs make little impact on the translation performance which is consistent with the discovery of Zhang et al. (2008a). However, if we append the non-contiguous phrasal rules derived from non-contiguous tree sequence pairs, no matter whether non-contiguous in source or in target, the performance statistically significantly ($p < 0.05$) improves (as presented in Exp 2~5), which validates our prediction that the non-contiguous rules derived from non-contiguous tree sequence pairs contribute more to the performance than those acquired from contiguous tree sequence pairs.

2) Not only that, after comparing Exp 6,7,8 against Exp 3,4,5 respectively, we find that the ability of rules derived from non-contiguous tree sequence pairs generally covers that of the rules derived from the contiguous tree sequence pairs, due to the slight change in BLEU score.

3) The further comparison of the non-contiguous rules from non-contiguous spans in Exp. 6&7 as well as Exp 3&4, shows that non-contiguity in the target side in Chinese-English translation task is not so useful as that in the source side when constructing the non-contiguous phrasal rules. This also validates the findings in Wellington et al. (2006) that varying the gaps on the English side (the target side in this context) seldom reduce the hierarchical alignment failures.

Table 3 explores the contribution of the non-contiguous translational equivalence to phrase-based models (all the rules in Table 3 has no grammar tags, but a gap <***> is allowed in the last three rows). **tgtncBP** refers to the bilingual phrases with gaps in the target side; **srcncBP** refers to the bilingual phrases with gaps in the source side; **src&tgtncBP** refers to the bilingual phrases with gaps in either side.

System	Rule Set	BLEU
Moses	cBP	23.86
Pisces	cBP	22.63
	cBP + tgtncBP	23.74
	cBP + srcncBP	23.93
	cBP + src&tgtncBP	24.24

Table 3: Performance of bilingual phrasal rules

1) As presented in Table 3, the effectiveness of the bilingual phrases derived from non-contiguous tree sequence pairs is clearly indicated. Models adopting both **tgtncBP** and **srcncBP** significantly ($p < 0.05$) outperform the model adopting **cBP** only.

2) Pisces underperforms Moses when utilizing **cBP**s only, since Pisces can only perform monotonic search with **cBP**s.

3) The bilingual phrase model with both **tgtncBP** and **srcncBP** even outperforms Moses. Compared with Moses, we only utilize plain features in Pisces for the bilingual phrase model (Feature 1~5 for all phrases and additional 7, 8 only for non-contiguous bilingual phrases as stated in Section 2.2; None of the complex reordering features or distortion features are employed by Pisces while Moses uses them), which suggests the effectiveness of the non-contiguous rules and the advantages of the proposed decoding algorithm.

Table 4 studies the impact on performance when setting different maximal gaps allowed for either side in a tree sequence pair (parameter g) and the relation with the quantity of rule set.

Significant improvement is achieved when allowing at least one gap on either side compared with when only allowing contiguous tree sequence pairs. However, the further increment of gaps does not benefit much. The result exhibits the accordance with the growing amplitude of the rule set filtered for the test set, in which the rule size increases more slowly as the maximal number of gaps increments. As a result, this slow increase against the increment of gaps can be probably attributed to the small augmentation of the effective

Max gaps allowed		Rule #	BLEU
source	target		
0	0	1,661,045	25.92
1	1	+841,263	26.53
2	2	+447,161	26.55
3	3	+17,782	26.56
∞	∞	+8,223	26.57

Table 4: Performance and rule size changing with different maximal number of gaps

	Output & References
Source	才/only 过/pass 了/null 五年/five years， 两人/two people 就/null 对簿公堂/confront at court
Reference	after only five years the two confronted each other at court
STSSG	*only* in the *five years* , *the two* candidates would 对簿公堂
SncTSSG key rules	*the two* people can confront *other* countries *at court* leisurely manner *only* in the *five years*
	VV(对簿公堂)→VB(confront)NP(JJ(other),NNS(countries))IN(at) NN(court) *** JJ(leisurely)NN(manner)
Source	欧元/Euro 的/'s 大幅/substantial 升值/appreciation 将/will 在/in 近期/recent 的/'s 调查/survey 中/middle 持续/continue 对/for 经济/economy 信心/confidence 产生/produce 影响/impact
Reference	substantial appreciation of the euro will continue to impact the economic confidence in the recent surveys
STSSG	*substantial appreciation of the euro* has continued *to* have an *impact* on *confidence* in the *economy* , *in the recent surveys will*
SncTSSG key rules	*substantial appreciation of the euro will continue in the recent surveys* have an *impact* on *economic confidence*
	AD(将) *** VV(持续) → VP(MD(will),VB(continue)) P(在) *** LC(中) → IN(in)

Table 5: Sample translations (tokens in italic match the reference provided)

non-contiguous rules.

In order to facilitate a better intuition to the ability of the SncTSSG based model against the STSSG based model, we present in Table 5, two translation outputs produced by both models.

In the first example, GIZA++ wrongly aligns the idiom word "对簿公堂/confront at court" to a non-contiguous phrase "*confront other countries at court, ***, leisurely manner*" in training, in which only the first constituent "*confront other countries at court*" is reasonable, indicated from the key rules of SncTSSG leant from the training set. The STSSG or any contiguous translational equivalence based model is unable to attain the corresponding target output for this idiom word via the non-contiguous word alignment and consider it as an out-of-vocabulary (OOV). On the contrary, the SncTSSG based model can capture the non-contiguous tree sequence pair consistent with the word alignment and further provide a reasonable target translation. It suggests that SncTSSG can easily capture the non-contiguous translational candidates while STSSG cannot. Besides, SncTSSG is less sensitive to the error of word alignment when extracting the translation candidates than the contiguous translational equivalence based models.

In the second example, "在/in 近期/recent 的/'s 调查/survey 中/middle" is correctly translated into "*in the recent surveys*" by both the STSSG and SncTSSG based models. This suggests that the short non-contiguous phrase "在/in *** 中/middle" is well handled by both models. Nevertheless, as for the one with a larger gap, "将/will *** 持续/continue" is correctly translated and well reordering into "*will continue*" by SncTSSG but failed by STSSG. Although the STSSG is theoretically able to capture this phrase from the contiguous tree sequence pair, the richer context in the gap as in this example, the more difficult STSSG can correctly translate the non-contiguous phrases. This exhibits the flexibility of SncTSSG to the rich context among the non-contiguous constituents.

6 Conclusions and Future Work

In this paper, we present a non-contiguous tree sequence alignment model based on SncTSSG to enhance the ability of non-contiguous phrase modeling and the reordering caused by non-contiguous constituents with large gaps. A three-phase decoding algorithm is developed to facilitate the usage of non-contiguous translational equivalences (tree sequence pairs in this work) which provides much flexibility for the reordering of the non-contiguous constituents with rich syntactic structural information. The experimental results show that our model outperforms the baseline models and verify the effectiveness of non-contiguous translational equivalences to non-contiguous phrase modeling in both syntax-based and phrase-based systems. We also find that in Chinese-English translation task, gaps are more effective in Chinese side than in the English side.

Although the characteristic of more sensitiveness to word alignment error enables SncTSSG to capture the additional non-contiguous language phenomenon, it also induces many redundant non-contiguous rules. Therefore, further work of our studies includes the optimization of the large rule set of the SncTSSG based model.

References

Rens Bod. 2007. Unsupervised Syntax-Based Machine Translation: The Contribution of Discontinuous Phrases. MT-Summmit-07. 51-56.

David Chiang. 2006. An Introduction to Synchronous Grammars. Tutorial on ACL-06

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insert grammars. ACL-05. 541-548

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. ACL-03.

Michel Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang and I. Thayer. 2006. Scalable Inference and training of context-rich syntactic translation models. COLING-ACL-06. 961-968

Daniel Gildea. 2003. Loosely Tree-Based Alignment for Machine Translation. ACL-03. 80-87.

Mary Hearne and Andy Way. 2003. Seeing the wood for the trees: data-oriented translation. MT Summit IX, 165-172.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. ACL-03. 423-430.

Philipp Koehn, Franz J. Och and Daniel Marcu. 2003. Statistical phrase-based translation. HLT-NAACL-03. 127-133

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. ACL-07. 77-180.

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

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

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. EMNLP-02, 133-139

Daniel Marcu, W. Wang, A. Echihabi and K. Knight. 2006. SPMT: statistical machine translation with syntactified target language phrases. EMNLP-06. 44-52.

Franz J. Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. Computational Linguistics, 30(4):417-449

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

Chris Quirk, Arul Menezes and Colin Cherry. 2005. Dependency treelet translation: syntactically informed phrasal SMT. ACL-05. 271-279.

S. Shieber. 2004. Synchronous grammars as tree transducers. In Proceedings of the Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms

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

Benjamin Wellington, Sonjia Waxmonsky and I. Dan Melamed. 2006. Empirical Lower Bounds on the Complexity of Translational Equivalence. ACL-06. 977-984

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. ACL-01. 523-530

Min Zhang, Hongfei Jiang, AiTi Aw, Jun Sun, Sheng Li and Chew Lim Tan. 2007. A tree-to-tree alignment-based model for statistical machine translation. MT-Summit-07. 535-542.

Min Zhang, Hongfei Jiang, AiTi Aw, Haizhou Li, Chew Lim Tan and Sheng Li. 2008a. A tree sequence alignment-based tree-to-tree translation model. ACL-08. 559-567.

Min Zhang, Hongfei Jiang, Haizhou Li, Aiti Aw, Sheng Li. 2008b. Grammar Comparison Study for Translational Equivalence Modeling and Statistical Machine Translation. COLING-08. 1097-1104.

Ying Zhang. Stephan Vogel. Alex Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? LREC-04. 2051-2054.