

Source-side Syntactic Reordering Patterns with Functional Words for Improved Phrase-based SMT

Jie Jiang, Jinhua Du, Andy Way

CNGL, School of Computing, Dublin City University

{jjiang, jdu, away}@computing.dcu.ie

Abstract

Inspired by previous source-side syntactic reordering methods for SMT, this paper focuses on using automatically learned syntactic reordering patterns with functional words which indicate structural reorderings between the source and target language. This approach takes advantage of phrase alignments and source-side parse trees for pattern extraction, and then filters out those patterns without functional words. Word lattices transformed by the generated patterns are fed into PB-SMT systems to incorporate potential reorderings from the inputs. Experiments are carried out on a medium-sized corpus for a Chinese–English SMT task. The proposed method outperforms the baseline system by 1.38% relative on a randomly selected testset and 10.45% relative on the NIST 2008 testset in terms of BLEU score. Furthermore, a system with just 61.88% of the patterns filtered by functional words obtains a comparable performance with the unfiltered one on the randomly selected testset, and achieves 1.74% relative improvements on the NIST 2008 testset.

1 Introduction

Previous work has shown that the problem of structural differences between language pairs in SMT can be alleviated by source-side syntactic reordering. Taking account for the integration with SMT systems, these methods can be divided into two different kinds of approaches (Elming,

2008): the deterministic reordering and the non-deterministic reordering approach.

To carry out the deterministic approach, syntactic reordering is performed uniformly on the training, devset and testset before being fed into the SMT systems, so that only the reordered source sentences are dealt with while building during the SMT system. In this case, most work is focused on methods to extract and to apply syntactic reordering patterns which come from manually created rules (Collins et al., 2005; Wang et al., 2007a), or via an automatic extraction process taking advantage of parse trees (Collins et al., 2005; Habash, 2007). Because reordered source sentence cannot be undone by the SMT decoders (Al-Onaizan et al., 2006), which implies a systematic error for this approach, classifiers (Chang et al., 2009b; Du & Way, 2010) are utilized to obtain high-performance reordering for some specialized syntactic structures (e.g. *DE* construction in Chinese).

On the other hand, the non-deterministic approach leaves the decisions to the decoders to choose appropriate source-side reorderings. This is more flexible because both the original and reordered source sentences are presented in the inputs. Word lattices generated from syntactic structures for N-gram-based SMT is presented in (Crego et al., 2007). In (Zhang et al., 2007a; Zhang et al., 2007b), chunks and POS tags are used to extract reordering rules, while the generated word lattices are weighted by language models and reordering models. Rules created from a syntactic parser are also utilized to form weighted n-best lists which are fed into the decoder (Li et al., 2007). Furthermore, (Elming, 2008; Elm-

ing, 2009) uses syntactic rules to score the output word order, both on English–Danish and English–Arabic tasks. Syntactic reordering information is also considered as an extra feature to improve PBSMT in (Chang et al., 2009b) for the Chinese–English task. These results confirmed the effectiveness of syntactic reorderings.

However, for the particular case of Chinese source inputs, although the *DE* construction has been addressed for both PBSMT and HPBSMT systems in (Chang et al., 2009b; Du & Way, 2010), as indicated by (Wang et al., 2007a), there are still lots of unexamined structures that imply source-side reordering, especially in the non-deterministic approach. As specified in (Xue, 2005), these include the *bei*-construction, *ba*-construction, three kinds of *de*-construction (including *DE* construction) and general preposition constructions. Such structures are referred with functional words in this paper, and all the constructions can be identified by their corresponding tags in the Penn Chinese TreeBank. It is interesting to investigate these functional words for the syntactic reordering task since most of them tend to produce structural reordering between the source and target sentences.

Another related work is to filter the bilingual phrase pairs with closed-class words (Sánchez-Martínez, 2009). By taking account of the word alignments and word types, the filtering process reduces the phrase tables by up to a third, but still provide a system with competitive performance compared to the baseline. Similarly, our idea is to use special type of words for the filtering purpose on the syntactic reordering patterns.

In this paper, our objective is to exploit these functional words for source-side syntactic reordering of Chinese–English SMT in the non-deterministic approach. Our assumption is that syntactic reordering patterns with functional words are the most effective ones, and others can be pruned for both speed and performance.

To validate this assumption, three systems are compared in this paper: a baseline PBSMT system, a syntactic reordering system with all patterns extracted from a corpus, and a syntactic reordering system with patterns filtered with functional words. To accomplish this, firstly the lat-

tice scoring approach (Jiang et al., 2010) is utilized to discover non-monotonic phrase alignments, and then syntactic reordering patterns are extracted from source-side parse trees. After that, functional word tags specified in (Xue, 2005) are adopted to perform pattern filtering. Finally, both the unfiltered pattern set and the filtered one are used to transform inputs into word lattices to present potential reorderings for improving PBSMT system. A comparison between the three systems is carried out to examine the performance of syntactic reordering as well as the usefulness of functional words for pattern filtering.

The rest of this paper is organized as follows: in section 2 we describe the extraction process of syntactic reordering patterns, including the lattice scoring approach and the extraction procedures. Then section 3 presents the filtering process used to obtain patterns with functional words. After that, section 4 shows the generation of word lattices with patterns, and experimental setup and results included related discussion are presented in section 5. Finally, we give our conclusion and avenues for future work in section 6.

2 Syntactic reordering patterns extraction

Instead of top-down approaches such as (Wang et al., 2007a; Chang et al., 2009a), we use a bottom-up approach similar to (Xia et al., 2004; Crego et al., 2007) to extract syntactic reordering patterns from non-monotonic phrase alignments and source-side parse trees. The following steps are carried out to extract syntactic reordering patterns: 1) the lattice scoring approach proposed in (Jiang et al., 2010) is used to obtain phrase alignments from the training corpus; 2) reordering regions from the non-monotonic phrase alignments are used to identify minimum treelets for pattern extraction; and 3) the treelets are transformed into syntactic reordering patterns which are then weighted by their occurrences in the training corpus. Details of each of these steps are presented in the rest of this section.

2.1 Lattice scoring for phrase alignments

The lattice scoring approach is proposed in (Jiang et al., 2010) for the SMT data cleaning task.

To clean the training corpus, word alignments are used to obtain approximate decoding results, which are then used to calculate BLEU (Papineni et al., 2002) scores to filter out low-scoring sentences pairs. The following steps are taken in the lattice scoring approach: 1) train an initial PBSMT model; 2) collect anchor pairs containing source and target phrase positions from word alignments generated in the training phase; 3) build source-side lattices from the anchor pairs and the translation model; 4) search on the source-side lattices to obtain approximate decoding results; 5) calculate BLEU scores for the purpose of data cleaning.

Note that the source-side lattices in step 3 come from anchor pairs, so each edge in the lattices contain both the source and target phrase positions. Thus the outputs of step 4 contain phrase alignments on the training corpus. These phrase alignments are used to identify non-monotonic areas for the extraction of reordering patterns.

2.2 Reordering patterns

Non-monotonic regions of the phrase alignments are examined as potential source-side reorderings. By taking a bottom-up approach, the reordering regions are identified and mapped to minimum treelets on the source parse trees. After that, syntactic reordering patterns are transformed from these minimum treelets.

In this paper, reordering regions A and B indicating swapping operations on the source side are only considered as potential source-side reorderings. Thus, given reordering regions AB , this implies (1):

$$AB \Rightarrow BA \quad (1)$$

on the source-side word sequences. Referring to the phrase alignment extraction in the last section, each non-monotonic phrase alignment produces one reordering region. Furthermore, for each reordering region identified, all of its sub-areas indicating non-monotonic alignments are also attempted to produce more reordering regions.

To represent the reordering region using syntactic structure, given the extracted reordering regions AB , the following steps are taken to map them onto the source-side parse trees, and to generate corresponding patterns:

1. Generate a parse tree for each of the source sentences. The Berkeley parser (Petrov, 2006) is used in this paper. To obtain simpler tree structures, right-binarization is performed on the parse trees, while tags generated from binarization are not distinguished from the original ones (e.g. $@VP$ and VP are the same).
2. Map reordering regions AB onto the parse trees. Denote N_A as the set of leaf nodes in region A and N_B for region B . The mapping is carried out on the parse tree to find a **minimum** treelet T , which satisfies the following two criteria: 1) there must exist a path from each node in $N_A \cup N_B$ to the root node of T ; 2) each leaf node of T can only be the ancestor of nodes in N_A or N_B (or none of them).
3. Traverse T in pre-order to obtain syntactic reordering pattern P . Label all the leaf nodes of T with A or B as reorder options, which indicate that the descendants of nodes with label A are supposed to be swapped with those with label B .

Instead of using *subtrees*, we use *treelets* to refer the located parse tree substructures, since treelets do not necessarily go down to leaf nodes.

Since phrase alignments cannot always be perfectly matched with parse trees, we also expand AB to the right and/or the left side with a limited number of words to find a minimum treelet. In this situation, a minimum number of ancestors of expanded tree nodes are kept in T but they are assigned the same labels as those from which they have been expanded. In this case, the expanded tree nodes are considered as the context nodes of syntactic reordering patterns.

Figure 1 illustrates the extraction process. Note the symbol $@$ indicates the right-binarization symbols (e.g. $@VP$ in the figure). In the figure, tree T (surrounded by dashed lines) is the minimum treelet mapped from the reordering region AB . Leaf node NP is labeled by A , VP is labeled by B , and the context node P is also labeled by A . Leaf nodes labeled A or B are collected into node sequences L_A or L_B to indicate the reordering op-

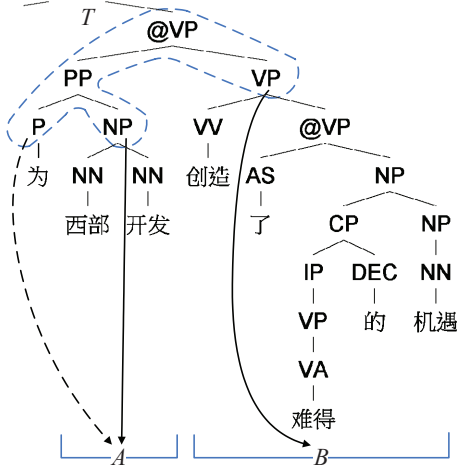


Figure 1: Reordering pattern extraction

erations. Thus the syntactic reordering pattern P is obtained from T as in (2):

$$P = \{VP (PP (P NP) VP) | O = \{L_A, L_B\}\} \quad (2)$$

where the first part of P is the VP with its tree structure, and the second part O indicates the reordering scheme, which implies that source words corresponding with descendants of L_A are supposed to be swapped with those of L_B .

2.3 Pattern weights estimation

We use p_{reo} to represent the chance of reordering when a treelet is located by a pattern on the parse tree. It is estimated by the number of reorderings for each of the occurrences of the pattern as in (3):

$$p_{reo}(P) = \frac{\text{count}\{\text{reorderings of } P\}}{\text{count}\{\text{observation of } P\}} \quad (3)$$

By contrast, one syntactic pattern P usually contains several reordering schemes (specified in formula (2)), each of them weighted as in (4):

$$w(O, P) = \frac{\text{count}\{\text{reorderings of } O \text{ in } P\}}{\text{count}\{\text{reorderings of } P\}} \quad (4)$$

Generally, a syntactic reordering pattern is expressed as in (5):

$$P = \{\text{tree} \mid p_{reo} \mid O_1, w_1, \dots, O_n, w_n\} \quad (5)$$

where $tree$ is the tree structures of the pattern, p_{reo} is the reordering probability, O_i and w_i are the reordering schemes and weights ($1 \leq i \leq n$).

3 Patterns with functional words

Some of the patterns extracted may not benefit the final system since the extraction process is controlled by phrase alignments rather than syntactic knowledge. Inspired by the study of *DE* constructions (Chang et al., 2009a; Du & Way, 2010), we assume that syntactic reorderings are indicated by functional words for the Chinese–English task. To incorporate the knowledge of functional words into the extracted patterns, instead of directly specifying the syntactic structure from the linguistic aspects, we use functional word tags to filter the extracted patterns. In this case, we assume that all patterns containing functional words tend to produce meaningful syntactic reorderings. Thus the filtered patterns carry the reordering information from the phrase alignments as well as the linguistic knowledge. Thus the noise produced in phrase alignments and the size of pattern set can be reduced, so that the speed and the performance of the system can be improved.

The functional word tags used in this paper are shown in Table 1, which come from (Xue, 2005). We choose them as functional words because normally they imply word reorderings between Chinese and English sentence pairs.

Tag	Description
BA	<i>ba</i> -construction
DEC	<i>de</i> (1 st kind) in a relative-clause
DEG	associative <i>de</i> (1 st kind)
DER	<i>de</i> (2 nd kind) in V- <i>de</i> const. & V- <i>de</i> -R
DEV	<i>de</i> (3 rd kind) before VP
LB	<i>bei</i> in long <i>bei</i> -construction
P	preposition excluding <i>bei</i> and <i>ba</i>
SB	<i>bei</i> in short <i>bei</i> -construction

Table 1: Syntactic reordering tags for functional words

Note that there are three kinds of *de*-constructions, but only the first kind is the *DE* construction in (Chang et al., 2009a; Du & Way, 2010). After the filtering process, both the unfiltered pattern set and the filtered one are used to build different syntactic reordering PBSMT systems for comparison purpose.

4 Word lattice construction

Both the devset and testset are transformed into word lattices by the extracted patterns to incorporate potential reorderings. Figure 2 illustrates this process: treelet T' is matched with a pattern, then its leaf nodes $\{a_1, \dots, a_m\} \in L_A$ (spanning $\{w_1, \dots, w_p\}$) are swapped with leaf nodes $\{b_1, \dots, b_n\} \in L_B$ (spanning $\{v_1, \dots, v_q\}$) on the generated paths in the word lattice.

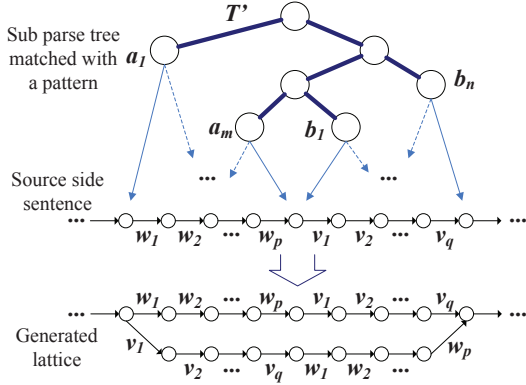


Figure 2: Incorporating potential reorderings into lattices

We sort the matched patterns by p_{reo} in formula (5), and only apply a pre-defined number of reorderings for each sentence. For each lattice node, if we denote E_0 as the edge from the original sentence, while patterns $\{P_1, \dots, P_i, \dots, P_k\}$ are applied to this node, then E_0 is weighted as in (6):

$$w(E_0) = \alpha + \sum_{i=1}^k \left\{ \frac{(1-\alpha)}{k} * \{1 - p_{reo}(P_i)\} \right\} \quad (6)$$

where $p_{reo}(P_i)$ is the pattern weight in formula (3), and α is the base probability to avoid E_0 being equal to zero. Suppose $\{E_s, \dots, E_{s+r-1}\}$ are generated by r reordering schemes of P_i , then E_j is weighted as in (7):

$$w(E_j) = \frac{(1-\alpha)}{k} * p_{reo}(P_i) * \frac{w_{s-j+1}(P_i)}{\sum_{t=1}^r w_t(P_i)} \quad (7)$$

where $w_t(P_i)$ is the reordering scheme in formula (5), and $s \leq j < s + r$. Reordering patterns with the same root lattice node share equal probabilities in formula (6) and (7).

5 Experiments and results

We conducted our experiments on a medium-sized corpus FBIS (a multilingual paragraph-aligned corpus with LDC resource number LDC2003E14) for the Chinese–English SMT task. The Champollion aligner (Ma, 2006) is utilized to perform sentence alignment. A total number of 256,911 sentence pairs are obtained, while 2,000 pairs for devset and 2,000 pairs for testset are randomly selected, which we call FBIS set. The rest of the data is used as the training corpus.

The baseline system is Moses (Koehn et al., 2007), and GIZA++¹ is used to perform word alignment. Minimum error rate training (MERT) (Och, 2003) is carried out for tuning. A 5-gram language model built via SRILM² is used for all the experiments in this paper.

Experiments results are reported on two different sets: the FBIS set and the NIST set. For the NIST set, the NIST 2005 testset (1,082 sentences) is used as the devset, and the NIST 2008 testset (1,357 sentences) is used as the testset. The FBIS set contains only one reference translation for both devset and testset, while NIST set has four references.

5.1 Pattern extraction and filtering with functional words

The lattice scoring approach is carried out with the same baseline system as specified above to produce the phrase alignments. The initial PB-SMT system in the lattice scoring approach is tuned with the FBIS devset to obtain the weights. As specified in section 2.1, phrase alignments are generated in the step 4 of the lattice scoring approach.

From the generated phrase alignments and source-side parse trees of the training corpus, we obtain 48,285 syntactic reordering patterns (57,861 reordering schemes) with an average number of 11.02 non-terminals. For computational efficiency, any patterns with number of non-terminal less than 3 and more than 9 are pruned. This procedure leaves 18,169 syntactic reordering patterns (22,850 reordering schemes) with a aver-

¹<http://fjoch.com/GIZA++.html>

²<http://www.speech.sri.com/projects/srilm/>

age number of 7.6 non-terminals. This pattern set is used to build the syntactic reordering PBSMT system without pattern filtering, which here after we call the ‘unfiltered system’.

Using the tags specified in Table 1, the extracted syntactic reordering patterns without functional words are filtered out, while only 6,926 syntactic reordering patterns (with 9,572 reordering schemes) are retained. Thus the pattern set are reduced by 61.88%, and over half of them are pruned by the functional word tags. The filtered pattern set is used to build the syntactic reordering PBSMT system with pattern filtering, which we refer as the ‘filtered system’.

Type	Tag	Patterns	Percent
<i>ba</i> -const.	BA	222	3.20%
<i>bei</i> -const.	LB	97	2.79%
	SB	96	
<i>de</i> -const. (1 st)	DEC	1662	60.11%
	DEG	2501	
<i>de</i> -const. (2 nd)	DER	52	0.75%
<i>de</i> -const. (3 rd)	DEV	178	2.57%
preposition excl. <i>ba</i> & <i>bei</i>	P	2591	37.41%

Table 2: Statistics on the number of patterns for each type of functional word

Statistics on the patterns with respect to functional word types are shown in Table 2. The number of patterns for each functional word in the filtered pattern set are illustrated, and percentages of functional word types are also reported. Note that some patterns contain more than one kind of functional word, so that the percentages of functional word types do not sum to one.

As demonstrated in Table 2, the first kind of *de*-construction takes up 60.11% of the filtered pattern set, and is the main type of patterns used in our experiment. This indicates that more than half of the patterns are closely related to the *DE* construction examined in (Chang et al., 2009b; Du & Way, 2010). However, the general preposition construction (excluding *bei* and *ba*) accounts for 37.41% of the filtered patterns, which implies that it is also a major source of syntactic reordering. By contrast, other constructions have much smaller amount of percentages, so have a minor

impact on our experiments.

5.2 Word lattice construction

As specified in section 4, for both unfiltered and the filtered systems, both the devset and testset are converted into word lattices with the unfiltered and filtered syntactic reordering patterns respectively. To avoid a dramatic increase in size of the lattices, the following constraints are applied: for each source sentence, the maximum number of reordering schemes is 30, and the maximum span of a pattern is 30.

For the lattice construction, the base probability in (6) and (7) is set to 0.05. The two syntactic reordering PBSMT systems also incorporate the built-in reordering models (distance-based and lexical reordering) of Moses, and their weights in the log-linear model are tuned with respect to the devsets.

The effects of the pattern filtering by functional words are also reported in Table 3. For both the FBIS and NIST sets, the average number of nodes in word lattices are illustrated before and after pattern filtering. From the table, it is clear that the pattern filtering procedure dramatically reduces the input size for the PBSMT system. The reduction is up to 37.99% for the NIST testset.

Data set	Unfiltered	Filtered	Reduced
FBIS dev	183.13	131.38	28.26%
FBIS test	183.68	136.56	25.65%
NIST dev	175.78	115.89	34.07%
NIST test	149.13	92.48	<u>37.99%</u>

Table 3: Comparison of the average number of nodes in word lattices

5.3 Results on FBIS set

Three systems are compared on the FBIS set: the baseline PBSMT system, and the syntactic reordering systems with and without pattern filtering. Since the built-in reordering models of Moses are enabled, several values of the distortion limit (DL) parameter are chosen to validate consistency. The evaluation results on the FBIS set are shown in Table 4.

As shown in Table 4, the syntactic reordering systems with and without pattern filtering outper-

System	DL	BLEU	NIST	METE
Baseline	0	22.32	6.45	52.51
	6	23.67	6.63	54.07
	10	24.52	6.66	54.04
	<u>12</u>	<u>24.57</u>	<u>6.69</u>	<u>54.31</u>
Unfiltered	0	23.92	6.60	54.30
	6	24.57	6.68	54.64
	<u>10</u>	<u>24.98</u>	<u>6.71</u>	<u>54.67</u>
	12	24.84	6.69	54.65
Filtered	0	23.71	6.60	54.11
	6	24.65	6.68	54.61
	<u>10</u>	24.87	<u>6.71</u>	<u>54.84</u>
	<u>12</u>	<u>24.91</u>	6.7	54.51

Table 4: Results on FBIS testset (DL = distortion limit, METE=METEOR)

form the baseline system for each of the distortion limit parameters in terms of the BLEU, NIST and METEOR scores (scores in bold face). By contrast, the filtered systems has a comparable performance with the unfiltered system: for some of the distortion limits, the filtered systems even outperforms the unfiltered system (scores in bold face, e.g. BLEU and NIST for DL=12, METEOR for DL=10).

The best performance of the baseline system is obtained with distortion limit 12 (underlined); the best performance of the unfiltered system is achieved with distortion limit 10 (underlined); while for the filtered system, the best BLEU score is accomplished with distortion limit 12 (underlined), and the best NIST and METEOR scores are shown with distortion limit 10 (underlined). Thus the unfiltered system outperforms the baseline by 0.41 (1.67% relative) BLEU points, 0.02 (0.30% relative) NIST points and 0.36 (0.66% relative) METEOR points. By contrast, the filtered system outperforms the baseline by 0.34 (1.38% relative) BLEU points, 0.02 (0.30% relative) NIST points and 0.53 (0.98% relative) METEOR points.

Compared with the unfiltered system, pattern filtering with functional words degrades performance by 0.07 (0.28% relative) in term of BLEU, but improves the system by 0.17 (0.31% relative) in term of METEOR, while the two systems achieved the same best NIST score.

These results indicates that the filtered system has a comparable performance with the unfiltered one on the FBIS set, while both of them outperform the baseline.

5.4 Results on NIST set

The evaluation results on the NIST set are illustrated in Table 5.

System	DL	BLEU	NIST	METE
Baseline	0	14.43	5.75	45.03
	6	15.61	5.88	45.75
	10	15.73	5.78	45.27
	<u>12</u>	<u>15.89</u>	<u>6.16</u>	<u>45.88</u>
Unfiltered	0	16.77	6.54	47.16
	<u>6</u>	<u>17.25</u>	<u>6.67</u>	<u>47.65</u>
	<u>10</u>	<u>17.15</u>	<u>6.64</u>	<u>47.78</u>
	12	16.88	6.56	47.17
Filtered	0	16.79	6.64	47.67
	<u>6</u>	<u>17.55</u>	<u>6.71</u>	<u>48.06</u>
	<u>10</u>	<u>17.51</u>	<u>6.72</u>	<u>48.15</u>
	12	17.37	6.72	48.08

Table 5: Results on NIST testset (DL = distortion limit, METE=METEOR)

From Table 5, the unfiltered system outperforms the baseline system for each of the distortion limits in terms of the BLEU, NIST and METEOR scores (scores in bold face). By contrast, the filtered system also outperform the unfiltered system for each of the distortion limits in terms of the three evaluation methods (scores in bold face).

The best performance of the baseline system is obtained with distortion limit 12 (underlined), while the best performance of the unfiltered system is obtained with distortion limit 6 for BLEU and NIST, and 10 for METEOR (underlined). For the filtered system, the best BLEU score is shown with distortion limit 6, and the best NIST and METEOR scores are accomplished with distortion limit 10 (underlined). Thus the unfiltered system outperforms the baseline by 1.36 (8.56% relative) BLEU points, 0.51 (8.28% relative) NIST points and 1.90 (4.14% relative) METEOR points. By contrast, the filtered system outperforms the baseline by 1.66 (10.45% relative) BLEU points, 0.56 (9.52% relative) NIST points and 2.27 (4.95% relative) METEOR points.

Compared with the unfiltered system, patterns with functional words boost the performance by 0.30 (1.74% relative) in term of BLEU, 0.05 (0.75% relative) in term of NIST, and 0.37 (0.77% relative) in term of METEOR.

These results demonstrate that the pattern filtering improves the syntactic reordering system on the NIST set, while both of them significantly outperform the baseline.

5.5 Discussion

Experiments in the previous sections demonstrate that: 1) the two syntactic reordering systems improve the PBSMT system by providing potential reorderings obtained from phrase alignments and parse trees; 2) patterns with functional words play a major role in the syntactic reordering process, and filtering the patterns with functional words maintains or even improves the system performance for Chinese–English SMT task. Furthermore, as shown in the previous section, pattern filtering prunes the whole pattern set by 61.88% and also reduces the sizes of word lattices by up to 37.99%, thus the whole syntactic reordering procedure for the original inputs as well as the tuning/decoding steps are sped up dramatically, which make the proposed methods more useful in the real world, especially for online SMT systems.

From the statistics on the filtered pattern set in Table 2, we also argue that the first kind of *de*-construction and general preposition (excluding *bei* and *ba*) are the main sources of Chinese–English syntactic reordering. Previous work (Chang et al., 2009b; Du & Way, 2010) showed the advantages of dealing with the *DE* construction. In our experiments too, even though all the patterns are automatically extracted from phrase alignments, these two constructions still dominate the filtered pattern set. This result confirms the effectiveness of previous work on *DE* construction, and also highlights the importance of the general preposition construction in this task.

6 Conclusion and future work

Syntactic reordering patterns with functional words are examined in this paper. The aim is to exploit these functional words within the syntactic reordering patterns extracted from phrase align-

ments and parse trees. Three systems are compared: a baseline PBSMT system, a syntactic reordering system with all patterns extracted from a corpus and a syntactic reordering system with patterns filtered with functional words. Evaluation results on a medium-sized corpus showed that the two syntactic reordering systems consistently outperform the baseline system. The pattern filtering with functional words prunes 61.88% of patterns, but still maintains a comparable performance with the unfiltered one on the randomly select testset, and even obtains 1.74% relative improvement on the NIST 2008 testset.

In future work, the structures of patterns containing functional words will be investigated to obtain fine-grained analysis on such words in this task. Furthermore, experiments on larger corpora as well as on other language pairs will also be carried out to validation our method.

Acknowledgements

This research is supported by Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University. Thanks to Yanjun Ma for the sentence-aligned FBIS corpus.

References

- Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. *Coling-ACL 2006: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529-536, Sydney, Australia.
- Pi-Chuan Chang, Dan Jurafsky, and Christopher D. Manning. 2009a. Disambiguating DE for Chinese–English machine translation. *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 215-223, Athens, Greece.
- Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009b. Discriminative reordering with Chinese grammatical features. *Proceedings of SSST-3: Third Workshop on Syntax and Structure in Statistical Translation*, pages 51-59, Boulder, CO.
- Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. *ACL-2005: 43rd Annual meeting of the*

- Association for Computational Linguistics*, pages 531-540, University of Michigan, Ann Arbor, MI.
- Josep M. Crego, and José B. Mariño. 2007. Syntax-enhanced N-gram-based SMT. *MT Summit XI*, pages 111-118, Copenhagen, Denmark.
- Jinhua Du and Andy Way. 2010. The Impact of Source-Side Syntactic Reordering on Hierarchical Phrase-based SMT. *EAMT 2010: 14th Annual Conference of the European Association for Machine Translation*, Saint-Raphaël, France.
- Jakob Elming. 2008. Syntactic reordering integrated with phrase-based SMT. *Coling 2008: 22nd International Conference on Computational Linguistics, Proceedings of the conference*, pages 209-216, Manchester, UK.
- Jakob Elming, and Nizar Habash. 2009. Syntactic reordering for English-Arabic phrase-based machine translation. *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 69-77, Athens, Greece.
- Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. *MT Summit XI*, pages 215-222, Copenhagen, Denmark.
- Jie Jiang, Andy Way, Julie Carson-Berndsen. 2010. Lattice Score-Based Data Cleaning For Phrase-Based Statistical Machine Translation. *EAMT 2010: 14th Annual Conference of the European Association for Machine Translation*, Saint-Raphaël, France.
- 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 2007: proceedings of demo and poster sessions*, pp. 177-180, Prague, Czech Republic.
- Chi-Ho Li, Dongdong Zhang, Mu Li, Ming Zhou, Minghui Li, and Yi Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. *ACL 2007: proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 720-727, Prague, Czech Republic.
- Xiaoyi Ma. 2006. Champollion: A Robust Parallel Text Sentence Aligner. *LREC 2006: Fifth International Conference on Language Resources and Evaluation*, pp.489-492, Genova, Italy.
- Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *ACL-2003: 41st Annual meeting of the Association for Computational Linguistics*, pp. 160-167, Sapporo, Japan.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method For Automatic Evaluation of Machine Translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, pp.311-318, Philadelphia, PA.
- Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. *Coling-ACL 2006: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433-440, Sydney, Australia.
- Felipe Sánchez-Martínez and Andy Way. 2009. Marker-based filtering of bilingual phrase pairs for SMT. *EAMT-2009: Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, pages 144-151, Barcelona, Spain.
- Chao Wang, Michael Collins, and Philipp Koehn. 2007a. Chinese syntactic reordering for statistical machine translation. *EMNLP-CoNLL-2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 737-745, Prague, Czech Republic.
- Fei Xia, and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. *Coling 2004: 20th International Conference on Computational Linguistics*, pages 508-514, University of Geneva, Switzerland.
- Nianwen Xue, Fei Xia, Fu-dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2), pages 207-238.
- Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP*, pages 333-341, Suntec, Singapore.
- Yuqi Zhang, Richard Zens, and Hermann Ney. 2007a. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. *SSST, NAACL-HLT-2007 AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1-8, Rochester, NY.
- Yuqi Zhang, Richard Zens, and Hermann Ney. 2007b. Improved chunk-level reordering for statistical machine translation. *IWSLT 2007: International Workshop on Spoken Language Translation*, pages 21-28, Trento, Italy.