Extracting Hierarchical Rules from a Weighted Alignment Matrix

Zhaopeng Tu, Yang Liu, Qun Liu and **Shouxun Lin**
Key Lab. of Intelligent Info. Processing
Institute of Computing Technology
Chinese Academy of Sciences
P.O. Box 2704, Beijing 100190, China
{tuzhaopeng,yliu,liuqun,sxlin}@ict.ac.cn

Abstract

Word alignment is a fundamental step in machine translation. Current statistical machine translation systems suffer from a major drawback: they only extract rules from 1-best alignments, which adversely affects the rule sets quality due to alignment mistakes. To alleviate this problem, we extract hierarchical rules from *weighted alignment matrix* (Liu et al., 2009). Since the sub-phrase pairs would change the inside and outside areas in the weighted alignment matrix of the hierarchical rules, we propose a new algorithm to calculate the relative frequencies and lexical weights of hierarchical rules. To achieve a balance between rule table size and performance, we construct a scoring measure that incorporates both frequency and lexical weight to select the best target phrase for each source phrase. Experiments show that our approach improves BLEU score by ranging from 1.4 to 1.9 points over baseline for hierarchical phrase-based, and 1.4 to 1.5 points for tree-to-string model.

1 Introduction

Word alignment plays an important role in statistical machine translation (SMT). Most SMT systems, not only phrase-based models (Och and Ney, 2004; Koehn et al., 2003; Xiong et al., 2006), but also syntax-based models (Chiang, 2005; Liu et al., 2006; Galley et al., 2006; Huang et al., 2006; Shen et al., 2008), usually extract rules from word aligned corpora. However, these systems suffer from a major drawback: they only extract rules from 1-best alignments, which adversely affects the rule sets quality due to alignment mistakes.

Typically, syntax-based models are more sensitive to word alignments because they care about inside (i.e., subtracted phrases). Figure 1(a) shows an alignment of a sentence pair. Since there is a wrong link (*de*, *of*), we could not extract many useful hierarchical rules such as (*zhongguo* X_1 *jingji*, *China* X_1 *economy*).To alleviate this problem, a natural solution is to extract rules from n-best alignments (Venugopal et al., 2008).

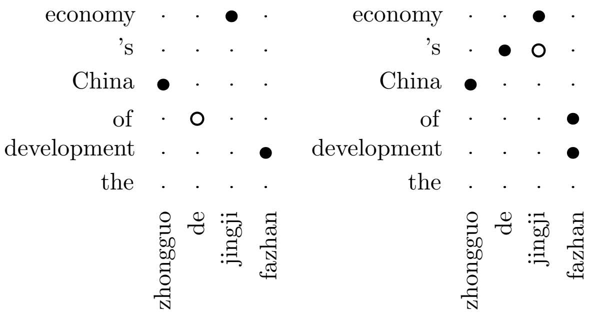


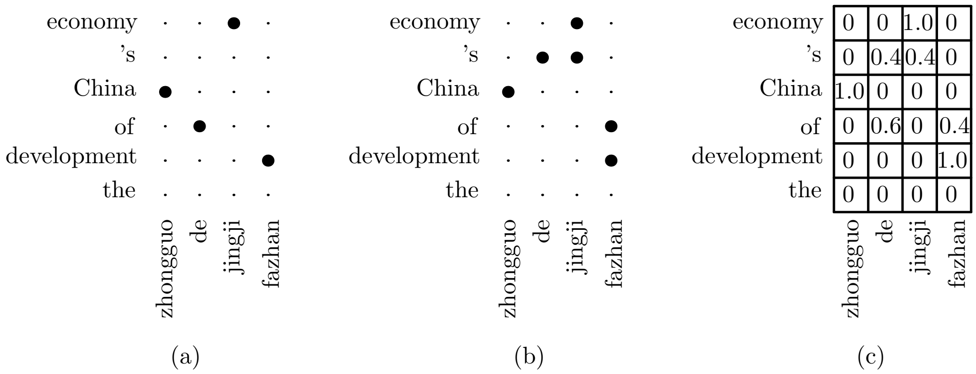
Figure 1: (a) One alignment of a sentence pair; (b) another alignment of the same sentence pair. Here coreless dots denote wrong links.

However, using n-best alignments still face two major challenges. First, n-best alignments have to be processed individually although they share many links, see (*zhongguo*, *China*) and (*jingji*, *economy*) in Figure 1. Second, regardless of probabilities of links in each alignment, numerous wrong rule would be extracted from n-best alignments. For example, a wrong rule (X_1 *de jingji*, *of* X_1 *'s economy*) would be extracted from the alignment in Figure 1(a).

Since Liu et al. (2009) show that weighted alignment matrix provides an elegant solution to these two drawbacks, we apply it to the hierarchical phrase-based model (Chiang, 2005) and the tree-to-string model (Liu et al., 2006; Huang et al., 2006). While such an idea seems intuitive, it is non-trivial to extract hierarchical rules from weighted alignment matrices.

Our work faces two major challenges. The first is how to calculate the relative frequencies and lex-

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1294–1303,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP



(a)

(b)

(c)

	zhongguo	de	jingji	fazhan
economy	0	0	1.0	0
's	0	0.4	0.4	0
China	1.0	0	0	0
of	0	0.6	0	0.4
development	0	0	0	1.0
the	0	0	0	0

Figure 2: (a) One alignment of a sentence pair; (b) another alignment of the same sentence pair; (c) the resulting weighted alignment matrix that samples the two alignments, of which the initial probabilities are 0.6 and 0.4 respectively.

ical weights of the rules with non-terminals (NTs). The sub-phrase pairs that are replaced with NTs in a rule, would change the inside and outside areas in the weighted alignment matrix of the rule. In addition, the sub-phrase pairs have their own probabilities and we should incorporate them to better estimate the probabilities of the hierarchical rules. Therefore, the calculations of relative frequencies and lexical weights for hierarchical rules are more complicated.

Another challenge is how to achieve a balance between performance and rule table size. Note that given a source phrase, there would be plenty of "potential" candidate target phrases in weighted matrices (Liu et al., 2009). If we retain all of them, these phrase pairs would produce even more hierarchical rules. For computational tractability, we need to design a measure to score the phrase pairs and wipe out the low-quality ones.

We propose a new algorithm to calculate the relative frequencies of rules, and construct a measure that incorporates both frequency and lexical weight to score target phrases. Experiments (Section 4) show that our approach improves BLEU score by ranging from 1.4 to 1.9 points over baseline for hierarchical phrase-based, and 1.4 to 1.8 points for tree-to-string model.

2 Weighted Alignment Matrix

A weighted alignment matrix (Liu et al., 2009) m is a $J \times I$ matrix to encode the probabilities of n-best alignments of the same sentence pair. Each element in the matrix stores a link probability $p_m(j,i)$, which is estimated from an n-best list by calculating relative frequencies:

$$p_m(j,i) \quad = \quad \frac{\sum_{a\in\mathcal{N}} p(a) \times \delta(a,j,i)}{\sum_{a\in\mathcal{N}} p(a)} \qquad (1)$$

where

$$\delta(a,j,i) = \begin{cases} 1 & (j,i) \in a \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Here $\mathcal{N}$ is an n-best list, $p(a)$ is the probability of an alignment a in the n-best list. The numbers in the cells in Figure 2(c) are the corresponding p_m.

Since $p_m(j,i)$ is the probability that f_j and e_i are aligned, the probability that the two words are not aligned is

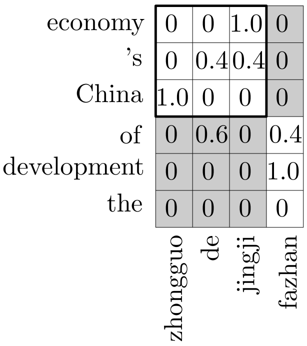
$$\bar{p}_m(j,i) = 1.0 - p_m(j,i) \qquad (3)$$

Figure 2 shows an example. The probability for the two words *zhongguo* and *China* being aligned is 1.0 and the probability that they are not aligned is 0.0. In another way, the two words are definitely aligned.

Given a phrase pair $(f_{j_1}^{j_2}, e_{i_1}^{i_2})$, Liu et al. (2009) calculate relative frequencies following Och and Ney (2004):

$$\phi(\tilde{e}|\tilde{f}) = \frac{count(f_{j_1}^{j_2}, e_{i_1}^{i_2})}{\sum_{e_{i'_1}^{i'_2}} count(f_{j_1}^{j_2}, e_{i'_1}^{i'_2})} \qquad (4)$$

The key point to calculate the relative frequency of the phrase pair is to obtain its fractional count. Liu et al. (2009) use the product of inside and outside probabilities as the fractional count of a phrase pair. Liu et al. (2009) define that inside probability indicates the probability that at least



	zhongguo	de	jingji	fazhan
economy	0	0	1.0	0
's	0	0.4	0.4	0
China	1.0	0	0	0
of	0	0.6	0	0.4
development	0	0	0	1.0
the	0	0	0	0

Figure 3: A weighted alignment matrix of a phrase pair. The light shading area is the outside area of phrase pair, and the area inside the pane with bold lines is the inside area.

one word in source phrase is aligned to a word in target phrase, and outside probability indicates the chance that no words in one phrase are aligned to a word outside the other phrase. The fractional count is calculated:

$$count(f_{j_1}^{j_2}, e_{i_1}^{i_2}) = \alpha(f_{j_1}^{j_2}, e_{i_1}^{i_2}) \times \beta(f_{j_1}^{j_2}, e_{i_1}^{i_2}) \quad (5)$$

where $\alpha(\cdot)$ and $\beta(\cdot)$ denote the inside and outside probabilities respectively, which can be calculated as

$$\alpha(\cdot) = 1 - \prod_{(j,i) \in in(\cdot)} \bar{p}_m(j,i) \quad (6)$$

$$\beta(\cdot) = \prod_{(j,i) \in out(\cdot)} \bar{p}_m(j,i) \quad (7)$$

Here $in(\cdot)$ denotes the inside area, which includes elements that fall inside the phrase pair, while $out(\cdot)$ denotes the outside area including elements that fall outside the phrase pair while fall in the same row or the same column. Figure 3 shows an example. The light shading area is the outside area of phrase pair and the area inside the pane with bold lines is the inside area.

To calculate the lexical weights, Liu et al. (2009) adapt $p_m(j,i)$ as the fractional count $count(f_j, e_i)$. The fractional counts of NULL words can be calculated as:

$$count(f_j, e_0) \quad = \quad \prod_{i=1}^{I} \bar{p}_m(j,i)$$

For example, in Figure 2, $count(de,'s)$ is 0.4 and $count(de, NULL)$ is 0.24.

Then the lexical weight can be calculated as:

$$p_w(\tilde{e}|\tilde{f}, m) = \prod_{i=1}^{|\tilde{e}|} \Bigg(\bigg(\frac{1}{\{j|p_m(j,i) > 0\}} \times \sum_{\forall j: p_m(j,i) > 0} p(e_i|f_j) \times p_m(j,i) \bigg) + p(e_i|f_0) \times \prod_{j=1}^{|\tilde{f}|} \bar{p}_m(j,i) \Bigg) \quad (8)$$

where

$$p(e_i|f_j) = \frac{count(f_j, e_i)}{\sum_{e'_i} count(f_j, e'_i)} \quad (9)$$

We apply weighted alignment matrix to the hierarchical phrase-based model (Chiang, 2007) and the tree-to-string model (Liu et al., 2006; Huang et al., 2006).

3 Rule Extraction

In hierarchical rules, both source and target sides are strings with NTs. In tree-to-string rules, the source side is a tree with NTs, while the target side is a string with NTs. Since the tree structure of source side has no effect on the calculations of relative frequencies and lexical weights, we can represent both tree-to-string and hierarchical rules as below:

$$X \to \langle \gamma, \alpha, \sim \rangle$$

where X is a nonterminal, γ and α are source and target strings (consist of terminals and NTs), and $\sim$ represents word alignments between NTs in γ and α.

The bulk of syntax grammars consists of two parts: *phrase pairs* and *variable rules*. The difference between them is containing NTs or not. Since we can calculate relative frequencies and lexical weights of phrase pairs as in Liu et al. (2009), we only focus on the calculation of variable rules.

3.1 Extraction Algorithm

Following Chiang (2007) and Liu et al. (2006), our extraction algorithm involves two steps. First, we extract phrase pairs from weighted alignment matrices. Then, we obtain variable rules by replacing sub-phrase pairs with NTs.

Figure 4 shows the algorithm of extracting phrase pairs from a weighted matrix for the hierarchical phrase-based model. The input of the algorithm is a sentence pair (f_1^J, e_1^I) that are both

```
1:  procedure PhraseExtraction($f_1^J$, $e_1^I$, $m$, $l$)
2:      $\mathcal{R} \leftarrow \emptyset$
3:      for $j_1 \leftarrow 1 \ldots J$ do
4:          $j_2 \leftarrow j_1$
5:          while $j_2 < J \wedge j_2 - j_1 < l$ do
6:              $T \leftarrow \{i | \exists j : j_1 \leq j \leq j_2 \wedge p_m(j, i) > 0\}$
7:              $i_l \leftarrow \textsc{min}(T)$
8:              $i_u \leftarrow \textsc{max}(T)$
9:              $r \leftarrow NULL$
10:             $s(r) \leftarrow -1$
11:             for $n \leftarrow 1 \ldots l$ do
12:                 for $i_1 \leftarrow i_l - n + 1 \ldots i_u$ do
13:                     $i_2 \leftarrow i_1 + n - 1$
14:                     if $s(f_{j_1}^{j_2}, e_{i_1}^{i_2}) > s(r)$ then
15:                         $r \leftarrow (f_{j_1}^{j_2}, e_{i_1}^{i_2})$
16:                         $s(r) \leftarrow s(f_{j_1}^{j_2}, e_{i_1}^{i_2})$
17:             $\mathcal{R} \leftarrow \mathcal{R} \cup \{r\}$
18:             $j_2 \leftarrow j_2 + 1$
19:     return $\mathcal{R}$
```

Figure 4: Algorithm of extracting phrase pairs from a sentence pair $\langle f_1^J, e_1^I \rangle$ annotated with a weighted alignment matrix m. We just retain the best target phrase for each source phrase. Here $s(\cdot)$ denotes the selection criteria in Section 3.2

target phrase	α	β	$count$
China 's economy	1.0	0.4	0.4
of China 's economy	1.0	0.6	0.6
China 's	1.0	0.0	0.0
of China 's	1.0	0.0	0.0

Table 1: Some candidate target phrases of the source phrase *zhongguo de jingji* in Figure 3 (suppose the structure of *zhongguo de jingji* is a complete sub-tree). Here α is inside probability, β is outside probability, and $count$ is fractional count.

strings, a weighted alignment matrix m, and a phrase length limit l. Note that we just retain the target phrase of highest score for each source phrase (lines 13-16). We describe these in Section 3.2. After we extract phrase pairs, we can obtain variable rules by replacing sub-phrase pairs with NTs.

We can also extend this algorithm to tree-to-string model. The difference is that the source sentence should be a tree instead of a string and additional syntactic constraints operate.

3.2 Selection Criteria

(Liu et al., 2009) show that given a source phrase, there would be multiple "potential" candidate target phrases in weighted matrices. Table 1 lists some candidate target phrases of the source phrase *zhongguo de jingji* in Figure 3. If we retain all of them, it will lead to an exponentially increasing rule table. To achieve balance between rule table size and performance, we just select the best candidate target phrase.

An interesting finding is that a target phrase with the largest fractional count is not always the best one. For example in Table 1, the target phrase *of China 's economy* has a larger fractional count than *China 's economy*. However, we can see that (*zhongguo de jingji*, *China 's economy*) is better.

To alleviate this problem, we incorporate lexical weight to distinguish good target phrases from bad ones. While frequency indicates how often the source phrase and target phrase occur together, lexical weight models the correspondence between them. Therefore, we can construct a scoring measure that incorporates both frequency and lexical weight. The scoring equation below models this effect:

$$s(\tilde{f}, \tilde{e}) = \omega \cdot count(\tilde{f}, \tilde{e}) + (1 - \omega) \cdot p_w(\tilde{e} | \tilde{f}, m) \quad (10)$$

where ω is the interpolation weight, $count(\tilde{f}, \tilde{e})$ is calculated by Equation 5, and $p_w(\tilde{e} | \tilde{f}, m)$ by Equation 8. In practice, we set $\omega = 0.5$.[1] Suppose $p_w($*China 's economy* $|$ *zhongguo de jingji*$)$ is 0.7 and $p_w($*of China 's economy* $|$ *zhongguo de jingji*$)$ is 0.4, then we should choose the target phrase *China 's economy* although *of China 's economy* has a larger fractional count.

Note that we select the best target phrase for each source phrase for just one sentence. It means there could still be many target phrases for each source phrase during decoding.

3.3 Calculating Relative Frequencies

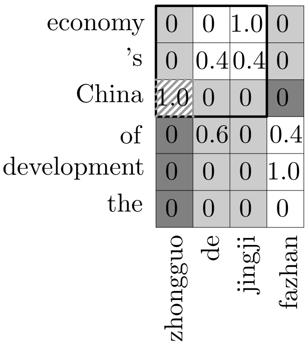
Figure 5 shows an example of the matrix of a hierarchical rule, which is generated from the phrase pair in Figure 3. Due to the existence of sub-phrase pairs, the inside and outside areas changes (see the difference between Figure 3 and Figure 5). Therefore, we can not simply calculate the outside probability of the hierarchical rule using the product of outside probabilities of phrase pair and sub-phrase pairs.

We follow Liu et al. (2009) to calculate relative frequencies using the product of inside and outside probabilities. We now extend the definitions of inside and outside probabilities to hierarchical rules that contain NTs.

[1] We tried a few other settings and found them to be less effective.

rule	α	β	$count$
X_1 *de jingji,* X_1 *'s economy*	1.0	0.4	0.4
zhongguo X_1, *China* X_1	1.0	0.4	0.4
zhongguo de X_1, *China 's* X_1	1.0	0.24	0.24
X_1 *de* X_2, X_1 *'s* X_2	1.0	0.24	0.24

Table 2: Some hierarchical rules generated from the phrase pair (*zhongguo de jingji*, *China's economy*) in Figure 3 (suppose the structure of *zhongguo de jingji* is a complete sub-tree). Here α is inside probability, β is outside probability, and $count$ is fractional count.



	zhongguo	de	jingji	fazhan
economy	0	0	1.0	0
's	0	0.4	0.4	0
China	1.0	0	0	0
of	0	0.6	0	0.4
development	0	0	0	1.0
the	0	0	0	0

Figure 5: A weighted alignment matrix of a variable rule, which is obtained by replacing (*zhongguo,China*) with X in (*zhongguo de jingji,China 's economy*). The diagonal area is the inside area of the sub-phrase pair. The shading area is the outside area of the variable rule, and heavy shading area is the duplicate outside area. The no shading area inside the pane with bold lines is the inside area.

Given a variable rule (f', e'), which is generated from the phrase pair ($f_{j_1}^{j_2}$, $e_{i_1}^{i_2}$) by replacing sub-phrase pairs with X. We denote R as the variable rule, P as the phrase pair ($f_{j_1}^{j_2}$, $e_{i_1}^{i_2}$), and X_k as the kth sub-phrase pair that is replaced with X. Therefore, the inside probability of a variable rule is calculated as:

$$\alpha(R) \quad = \quad \prod_k \alpha(X_k) \tag{11}$$

We tried to follow the constraints of Chiang (2007): (1) unaligned words are not allowed at the edges of phrases; (2) a rule must have at least one pair of aligned words. This would take into account the terminals in the variable rule, but make the calculation more complicated (especially constraint (1)). However, it didn't work well. Therefore, we only constraint that the rule should respect the word alignment, which means one terminal in a phrase could not align to another word outside the phrase (using outside probability).

Accordingly, the outside probability is calculated as:

$$\beta(R) = \prod_{(j,i)\in out(R)} \bar{p}_m(j,i) \tag{12}$$

where

$$out(R) = out(P) \bigcup \Big(\bigcup_k out(X_k) \Big)$$

For example, the inside probability of (X_1 *de jingji*, X_1 *'s economy*) in Figure 5 is 1.0, and its outside probability is 0.4.

We also use Equation 5 to calculate the fractional counts of hierarchical rules. We follow Liu et al. (2009) to prune rule table using a threshold of frequency. Table 2 lists some hierarchical rules generated from the phrase pair (*zhongguo de jingji, China's economy*) in Figure 3. If the threshold is 0.2, we retain all the rules in Table 2.

3.4 Calculating Lexical Weights

We denote S_R as all words in source side of the inside area of variable rule R, and T_R as the words in target side. For the rule (X_1 *de jingji*, X_1 *'s economy*) in Figure 5, S_R is {*de, jingji*} and T_R is {*'s*, *economy*}. Then, we can calculate the lexical weight as:

$$p_w(\tilde{e}|\tilde{f},m) = \prod_{i\in T_R} \Bigg(\bigg(\frac{1}{|\{j|p_m(j,i)>0\}|} \times \sum_{\forall j:p_m(j,i)>0} p(e_i|f_j) \times p_m(j,i) \bigg) + p(e_i|f_0) \times \prod_{j\in S_R} \bar{p}_m(j,i) \Bigg) \tag{13}$$

Note that we only consider each word pair (f_j, e_i) in the inside area of the variable rule. For example, the lexical weight of (X_1 *de jingji*, X_1 *'s* *economy*) is

$$\Big(\frac{1}{2} \times \big(p('s|de) \times 0.4 + p('s|jingji) \times 0.4\big) + p('s|NULL) \times 0.36\Big) \times \Big(p(economy|jingji) \times 1.0\Big)$$

Here the probability that *economy* translates a source NULL token is 0.0.

4 Experiments

4.1 Data Preparation

Our experiments are on Chinese-English translation based on replications of hierarchical phrase-based system (Chiang, 2007) and tree-to-string system (Liu et al., 2006). We train a 4-gram language model on the Xinhua portion of GIGAWORD corpus using the SRI Language Modeling Toolkit (Stolcke, 2002) with modified Kneser-Ney smoothing (Kneser and Ney, 1995). We optimize feature weights using the minimum error rate training algorithm (Och and Ney, 2002) on the NIST 2002 test set. We evaluate the translation quality using case-insensitive BLEU metric (Papineni et al., 2002) on the NIST 2003/2004/2005 test sets.

To obtain weighted alignment matrices, we follow Venugopal et al. (2008) to produce n-best lists via GIZA++. We produce 20-best lists in two translation directions, then used "grow-diag-final-and" (Koehn et al., 2003) to all 20×20 bidirectional alignment pairs. We follow Liu et al. (2009) to use $p_{s2t} \times p_{t2s}$ as the probabilities of an alignment pair. Analogously, we abandon duplicate alignments that are produced from different alignment pairs. After these steps, there are 110 candidate alignments on average for each sentence pair. We obtained n-best lists by selecting the top n alignments from 110-best lists. We re-estimated the probability of each alignment in the n-best list using re-normalization (Venugopal et al., 2008). Finally, we construct weighted alignment matrices from these n-best alignments.

We will first report results trained on a small-scaled corpus, and then scale to a larger one. When extracting tree-to-string rules, we limit the maximal height of rules to 3. We use the pruning threshold: $t = 0.5$.

4.2 Results on Small Data

To test the effect of our approach, we firstly carried out experiments on FBIS corpus, which contains 230K sentence pairs. Table 3 shows the rule table size and translation quality. Using n-best alignments slightly improved the BLEU score, but at the cost of much slower extraction, since each of top-n alignments has to be processed individually although they share many align links. Matrix-based extraction, by contrast, is much faster due to packing and produces consistently better BLEU scores. The absolute improvements of ranging from +1.6 to +1.8 BLEU points and +1.4 to +1.8 BLEU points over 1-best alignments for hierarchical phrase-based and tree-to-string models respectively, are statistically significant at $p < 0.01$ by using *sign-test* (Collins et al., 2005).

Basically, in the matrix case of the hierarchical phrase-based model, we can use about twice as many rules as in the 1-best case, or 1.3 times of 10-best extraction. However, in tree-to-string scenario, matrix-based extraction produces less rules than k-best extraction. We contribute this to the extra complete sub-tree constraint.

4.3 Results on Large Data

We also conducted experiments on a larger training data, which contains 1.5M sentence pairs coming from LDC dataset.[2]

The ruletable size and BLEU score are shown in Table 4. An interesting finding is that BLEU scores decline when using k-best extraction in some cases. We conjecture that some low-quality rules that harm the performance of decoder, are extracted from k-best alignments. Using weighted matrices on larger corpus also achieved significant and consistent improvements over using 1-best and n-best lists. These results confirm that our approach is a promising direction for syntax-based machine translation.

4.4 Comparison of Parameter Estimation

In this section we investigated the question of how many rules are shared by n-best and matrix-based extractions on small data (FBIS corpus). Our motivation is that weighted alignment matrices have been reported to be beneficial for better estimation of rule translation probabilities and lexical weights

[2] The corpus includes LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

Rules from...	Extraction	Total Rules	NIST03		NIST04		NIST05	
			Rules	BLEU	Rules	BLEU	Rules	BLEU
hierarchical phrase-based model								
1-best	17	39.7M	2.5M	30.14	4.2M	33.82	3.0M	30.33
10-best	155	62.7M	4.2M	30.59	7.0M	34.30	5.0M	30.73
m(10)	89	86.7M	5.7M	**31.81**	9.5M	**35.67**	6.6M	**31.94**
tree-to-string model								
1-best	21	9.3M	532K	27.39	762K	30.30	614K	27.06
10-best	231	19.6M	890K	27.57	1.13M	30.65	1.02M	27.07
m(10)	44	9.2M	590K	**28.92**	836K	**31.77**	677K	**28.87**

Table 3: Results with different rule extraction methods on small data. Here 1-best, 10-best and m(10) denote 1-best alignments, 10-best lists and weighted matrices estimated from 10-best lists respectively. The rules are filtered on the corresponding test set. "Extraction" denotes extraction time in millsecs per sentence pair. We evaluate the translation quality using 4-grams case-insensitive BLEU metric.

Rules from...	Total Rules	NIST03		NIST04		NIST05	
		Rules	BLEU	Rules	BLEU	Rules	BLEU
hierarchical phrase-based model							
1-best	204M	10.3M	33.40	16.1M	34.65	11.7M	32.88
10-best	288M	16.5M	33.18	25.2M	34.75	18.6M	32.47
m(10)	524M	26.1M	**35.10**	40.7M	**36.56**	29.5M	**34.31**
tree-to-string model							
1-best	30.7M	1.99M	30.76	2.68M	32.69	2.21M	30.36
10-best	71.4M	3.53M	31.54	4.63M	33.47	3.89M	31.09
m(10)	30.7M	2.24M	**32.23**	2.99M	**34.24**	2.48M	**31.88**

Table 4: Results with different rule extraction methods on large data. We use m(10) for the weighted matrices estimated from 10-best lists.

(Liu et al., 2009). The experiments are tested on NIST 2005 dataset.

Table 5 gives some statistics. We use $m(10)$ for the weighted matrices estimated from 10-best lists. "All" denotes the full rule table, "Shared" denotes the intersection of two tables, and "Non-shared" denotes the complement. There were 18.8% of rules learned from weighted matrices included by both tables in hierarchical phrase-based case, while 36.5% for tree-to-string rules, indicating that complete sub-tree constraint played an important role in matrix-based tree-to-string rule extraction. Note that the probabilities of "Shared" rules are different for the two approaches. Liu et al. (2009) shows that using matrices outperformed using n-best lists even with the same rules. Our experiments confirmed these findings.

4.5 Best Rule or More Rules

Someone would argue that using more rules could improve the performance, especially for the tree-to-string model. Therefore, we carried out experiments on small data for tree-to-string model to investigate which one is better. Note that even though we retain the best target side for each source side for each sentence, there could still be many target sides for each source side when decoding.

Table 6 shows the results of different criterions. The first column "Criteria" indicates how many target phrases are preserved: the best one or all phrases that reach pruning threshold. We can see that "More Rules" could not outperform "Best Rule" even using almost 2.5 times rules. One possible reason is that it might introduce some low-quality target phrase such as *of China 's economy* in Table 1, which will generate more substandard variable rules.

5 Related Works

Recent works have shown that machine translation can benefit when offered more alternatives. Mi

Rules from…	Shared		Non-shared		All	
	Rules	BLEU	Rules	BLEU	Rules	BLEU
hierarchical phrase-based model						
10-best	1.56M	28.42	4.66M	18.60	6.22M	30.73
$m(10)$	1.56M	29.07	6.89M	22.90	8.45M	31.94
tree-to-string model						
10-best	311K	23.00	707K	10.94	1018K	27.07
$m(10)$	311K	23.55	366K	11.92	677K	28.87

Table 5: Comparison of rule tables learned from n-best lists and weighted matrices. The rules are filtered on both development and test sets. "All" denotes the full rule table, "Shared" denotes the intersection of two tables, and "Non-shared" denotes the complement. Note that the probabilities of "Shared" rules are different for the two approaches.

Criteria	Total Rules	NIST03		NIST04		NIST05	
		Rules	BLEU	Rules	BLEU	Rules	BLEU
Best Rule	9.2M	590K	28.92	836K	31.77	677K	28.87
More Rules	21.4M	1.54M	29.07	1.97M	31.66	1.72M	29.02

Table 6: Comparison of rule tables learned from weighted matrices using different criterions. "Best Rule" denotes the rule table using the criteria described in Section 3.2, "More Rules" denotes the rule table using the criteria that retains all candidate target phrases that reach pruning threshold.

and Huang (2008) and Tu et al. (2010) use forests instead of 1-best trees; Venugopal et al. (2003) and Deng et al. (2008) soft the alignment consistency constraint to extract more rules; Dyer et al. (2008) use word lattices instead of 1-best segmentations to generate more alignments for a sentence pair; Venugopal et al. (2008) use n-best alignments directly for rule extraction.

To generate larger rule sets, de Gispert et al. (2010) extract hierarchical rules from alignment posterior probabilities. They concern how to extract larger rule sets using simple yet powerful hierarchical grammar, while we focus on whether weighted alignment matrix could overcome the alignment errors for different translation models (e.g. phrase-based, hierarchical phrase-based and tree-based models). They use phrase posteriors as the fractional count, while we use the product of inside and outside probabilities. Besides, they filter rules after extracting all rules from corpus, while we prune rules when extracting.

6 Conclusion and Future Works

Liu et al. (2009) proposed a new structure named weighted alignment matrix that make a better use of noisy alignments. Since weighted matrices proves effective for phrase-based model, we apply it to syntax-based models, which are more sensitive to word alignments. Due to the difference in structure between phrases and hierarchical rule, we develop new algorithms to calculate relative frequencies and lexical weights of hierarchical rules. To achieve a balance between rule table size and performance, we develop a scoring measure that incorporates both frequency and lexical weight to select the best target phrase for each source phrase. Our experiments show that our approach improves BLEU score significantly, with reasonable extraction speed, indicating that weighted alignment matrix also works for syntax-based models.

Besides the hierarchical phrase-based model and tree-to-string model, our method is also applicable to other paradigms such as the string-to-tree models (Galley et al., 2006) and the string-to-dependency models (Shen et al., 2008). Another interesting direction is to use a simpler alignment model that can compute alignment point posteriors directly, such as word-based ITG model (Zhang and Gildea, 2005; Haghighi et al., 2009).

7 Acknowledgement

The authors were supported by National Natural Science Foundation of China Contract 60736014 and 60903138. We thank the anonymous reviewers for their insightful comments.

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.

M. Collins, P. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, page 540. Association for Computational Linguistics.

Adrià de Gispert, Juan Pino, and William Byrne. 2010. Hierarchical phrase-based translation grammars extracted from alignment posterior probabilities. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 545–554, Cambridge, MA, October. Association for Computational Linguistics.

Yonggang Deng, Jia Xu, and Yuqing Gao. 2008. Phrase table training for precision and recall: what makes a good phrase and a good phrase pair? In *Proceedings of ACL-08: HLT*, pages 81–88, Columbus, Ohio, June. Association for Computational Linguistics.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL-08: HLT*, pages 1012–1020, Columbus, Ohio, June. Association for Computational Linguistics.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July. Association for Computational Linguistics.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised itg models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 923–931, Suntec, Singapore, August. Association for Computational Linguistics.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of AMTA*, pages 66–73. Citeseer.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *ICASSP IEEE INT CONF ACOUST SPEECH SIGNAL PROCESS PROC*, volume 1, pages 181–184.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July. Association for Computational Linguistics.

Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. 2009. Weighted alignment matrices for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1017–1026, Singapore, August. Association for Computational Linguistics.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 206–214, Honolulu, Hawaii, October. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

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, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of Seventh International Conference on Spoken Language Processing*, volume 3, pages 901–904. Citeseer.

Zhaopeng Tu, Yang Liu, Young-Sook Hwang, Qun Liu, and Shouxun Lin. 2010. Dependency forest for statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1092–1100, Beijing, China, August. Coling 2010 Organizing Committee.

Ashish Venugopal, Stephan Vogel, and Alex Waibel. 2003. Effective phrase translation extraction from alignment models. In *Proceedings of ACL*.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2008. Wider pipelines: n-best alignments and parses in mt training. In *Proceedings of AMTA*, Honolulu, Hawaii.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, Sydney, Australia, July. Association for Computational Linguistics.

Hao Zhang and Daniel Gildea. 2005. Stochastic lexicalized inversion transduction grammar for alignment. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 475–482. Association for Computational Linguistics.