Effective Use of Function Words for Rule Generalization in Forest-Based Translation

Xianchao Wu[†] **Takuya Matsuzaki**[†] **Jun'ichi Tsujii**[†‡*]

[†]Department of Computer Science, The University of Tokyo
[‡]School of Computer Science, University of Manchester
[*]National Centre for Text Mining (NaCTeM)
{wxc, matuzaki, tsujii}@is.s.u-tokyo.ac.jp

Abstract

In the present paper, we propose the effective usage of function words to generate generalized translation rules for forest-based translation. Given aligned forest-string pairs, we extract composed tree-to-string translation rules that account for multiple interpretations of both aligned and unaligned target function words. In order to constrain the exhaustive attachments of function words, we limit to bind them to the nearby syntactic chunks yielded by a target dependency parser. Therefore, the proposed approach can not only capture source-tree-to-target-chunk correspondences but can also use forest structures that compactly encode an exponential number of parse trees to properly generate target function words during decoding. Extensive experiments involving large-scale English-to-Japanese translation revealed a significant improvement of 1.8 points in BLEU score, as compared with a strong forest-to-string baseline system.

1 Introduction

Rule generalization remains a key challenge for current syntax-based statistical machine translation (SMT) systems. On the one hand, there is a tendency to integrate richer syntactic information into a translation rule in order to better express the translation phenomena. Thus, flat phrases (Koehn et al., 2003), hierarchical phrases (Chiang, 2005), and syntactic tree fragments (Galley et al., 2006; Mi and Huang, 2008; Wu et al., 2010) are gradually used in SMT. On the other hand, the use of syntactic phrases continues due to the requirement for phrase coverage in most syntax-based systems. For example, Mi et al. (2008) achieved a 3.1-point improvement in BLEU score (Papineni et al., 2002) by including bilingual syntactic phrases in their forest-based system. Compared with flat phrases, syntactic rules are good at capturing global reordering, which has been reported to be essential for translating between languages with substantial structural differences, such as English and Japanese, which is a subject-object-verb language (Xu et al., 2009).

Forest-based translation frameworks, which make use of packed parse forests on the source and/or target language side(s), are an increasingly promising approach to syntax-based SMT, being both algorithmically appealing (Mi et al., 2008) and empirically successful (Mi and Huang, 2008; Liu et al., 2009). However, forest-based translation systems, and, in general, most linguistically syntax-based SMT systems (Galley et al., 2004; Galley et al., 2006; Liu et al., 2006; Zhang et al., 2007; Mi et al., 2008; Liu et al., 2009; Chiang, 2010), are built upon word aligned parallel sentences and thus share a critical dependence on word alignments. For example, even a single spurious word alignment can invalidate a large number of otherwise extractable rules, and unaligned words can result in an exponentially large set of extractable rules for the interpretation of these unaligned words (Galley et al., 2006).

What makes word alignment so fragile? In order to investigate this problem, we manually analyzed the alignments of the first 100 parallel sentences in our English-Japanese training data (to be shown in Table 2). The alignments were generated by running GIZA++ (Och and Ney, 2003) and the *grow-diag-final-and* symmetrizing strategy (Koehn et al., 2007) on the training set. Of the 1,324 word alignment pairs, there were 309 error pairs, among

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 22–31,
Portland, Oregon, June 19-24, 2011.

which there were 237 target function words, which account for 76.7% of the error pairs[1]. This indicates that the alignments of the function words are more easily to be mistaken than content words. Moreover, we found that most Japanese function words tend to align to a few English words such as 'of' and 'the', which may appear anywhere in an English sentence. Following these problematic alignments, we are forced to make use of relatively large English tree fragments to construct translation rules that tend to be ill-formed and less generalized.

This is the motivation of the present approach of re-aligning the target function words to source tree fragments, so that the influence of incorrect alignments is reduced and the function words can be generated by tree fragments on the fly. However, the current dominant research only uses 1-best trees for syntactic realignment (Galley et al., 2006; May and Knight, 2007; Wang et al., 2010), which adversely affects the rule set quality due to parsing errors. Therefore, we realign target function words to a packed forest that compactly encodes exponentially many parses. Given aligned forest-string pairs, we extract composed tree-to-string translation rules that account for multiple interpretations of both aligned and unaligned target function words. In order to constrain the exhaustive attachments of function words, we further limit the function words to bind to their surrounding chunks yielded by a dependency parser. Using the composed rules of the present study in a baseline forest-to-string translation system results in a 1.8-point improvement in the BLEU score for large-scale English-to-Japanese translation.

2 Backgrounds

2.1 Japanese function words

In the present paper, we limit our discussion on Japanese particles and auxiliary verbs (Martin, 1975). Particles are suffixes or tokens in Japanese grammar that immediately follow modified content words or sentences. There are eight types of Japanese function words, which are classified depending on what function they serve: case markers, parallel markers, sentence ending particles, interjectory particles, adverbial particles, binding particles, conjunctive particles, and phrasal particles.

Japanese grammar also uses auxiliary verbs to give further semantic or syntactic information about the preceding main or full verb. Alike English, the extra meaning provided by a Japanese auxiliary verb alters the basic meaning of the main verb so that the main verb has one or more of the following functions: passive voice, progressive aspect, perfect aspect, modality, dummy, or emphasis.

2.2 HPSG forests

Following our precious work (Wu et al., 2010), we use head-drive phrase structure grammar (HPSG) forests generated by Enju[2] (Miyao and Tsujii, 2008), which is a state-of-the-art HPSG parser for English. HPSG (Pollard and Sag, 1994; Sag et al., 2003) is a lexicalist grammar framework. In HPSG, linguistic entities such as words and phrases are represented by a data structure called a *sign*. A sign gives a factored representation of the syntactic features of a word/phrase, as well as a representation of their semantic content. Phrases and words represented by signs are collected into larger phrases by the applications of *schemata*. The semantic representation of the new phrase is calculated at the same time. As such, an HPSG parse forest can be considered to be a forest of signs. Making use of these signs instead of part-of-speech (POS)/phrasal tags in PCFG results in a fine-grained rule set integrated with deep syntactic information.

For example, an aligned HPSG forest[3]-string pair is shown in Figure 1. For simplicity, we only draw the identifiers for the signs of the nodes in the HPSG forest. Note that the identifiers that start with 'c' denote non-terminal nodes (e.g., c0, c1), and the identifiers that start with 't' denote terminal nodes (e.g., t3, t1). In a complete HPSG forest given in (Wu et al., 2010), the terminal signs include features such as the POS tag, the tense, the auxiliary, the voice of a verb, etc.. The non-terminal signs include features such as the phrasal category, the name of the schema

[1]These numbers are language/corpus-dependent and are not necessarily to be taken as a general reflection of the overall quality of the word alignments for arbitrary language pairs.

[2]http://www-tsujii.is.s.u-tokyo.ac.jp/enju/index.html

[3]The forest includes three parse trees rooted at c0, c1, and c2. In the 1-best tree, 'by' modifies the passive verb 'verified'. Yet in the 2- and 3-best tree, 'by' modifies 'this result was verified'. Furthermore, 'verified' is an adjective in the 2-best tree and a passive verb in the 3-best tree.

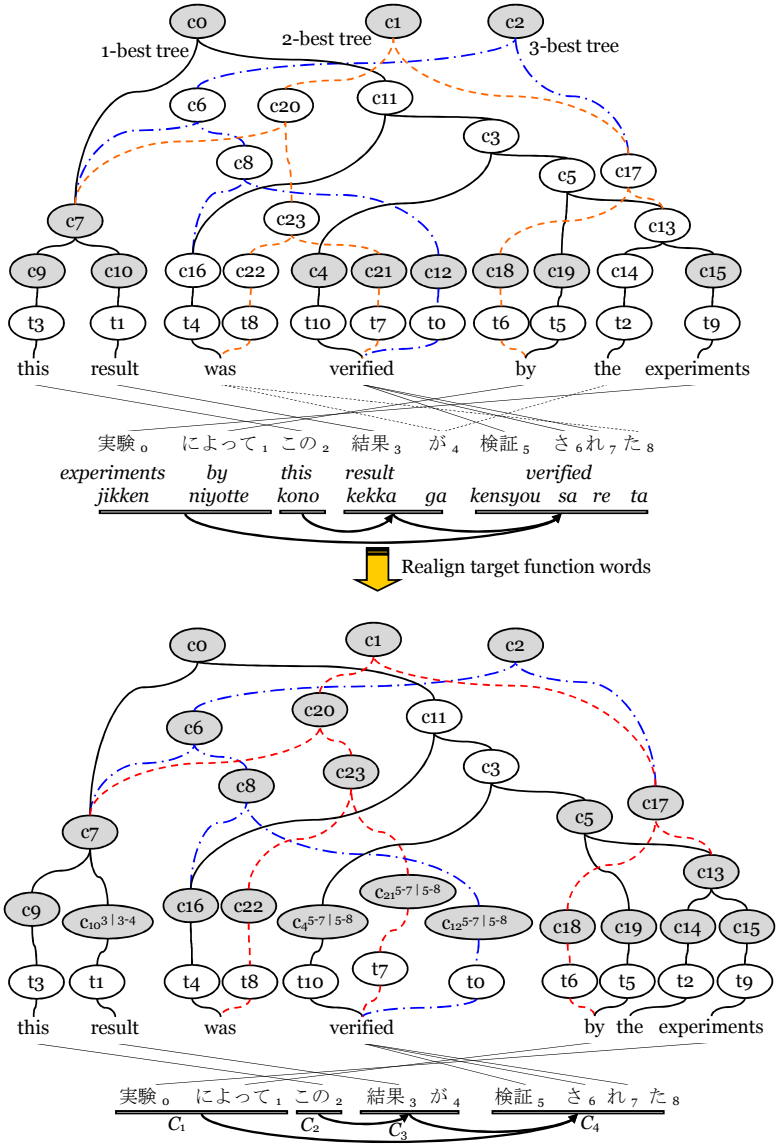

Figure 1: Illustration of an aligned HPSG forest-string pair for English-to-Japanese translation. The chunk-level dependency tree for the Japanese sentence is shown as well.

applied in the node, etc..

3 Composed Rule Extraction

In this section, we first describe an algorithm that attaches function words to a packed forest guided by target chunk information. That is, given a triple $\langle F_S, T, A \rangle$, namely an aligned ($A$) source forest ($F_S$) to target sentence ($T$) pair, we 1) tailor the alignment A by removing the alignments for target function words, 2) seek attachable nodes in the source forest F_S for each function word, and 3) construct a derivation forest by topologically traversing F_S. Then, we identify minimal and composed rules from the derivation forest and estimate the probabilities of rules and scores of derivations using the expectation-maximization (EM) (Dempster et al., 1977) algorithm.

3.1 Definitions

In the proposed algorithm, we make use of the following definitions, which are similar to those described in (Galley et al., 2004; Mi and Huang, 2008):

- $s(\cdot)$: the *span* of a (source) node v or a (target) chunk $\mathcal{C}$, which is an index set of the words that

v or $\mathcal{C}$ covers;

- $t(v)$: the *corresponding span* of v, which is an index set of aligned words on another side;
- $c(v)$: the *complement span* of v, which is the union of corresponding spans of nodes v' that share an identical parse tree with v but are neither antecedents nor descendants of v;
- $\mathcal{P}_A$: the *frontier set* of F_S, which contains nodes that are *consistent* with an alignment A (gray nodes in Figure 1), i.e., $t(v) \neq \varnothing$ and $closure(t(v)) \cap c(v) = \varnothing$.

The function $closure$ covers the gap(s) that may appear in the interval parameter. For example, $closure(t(\text{c3})) = closure(\{\text{0-1, 4-7}\}) = \{\text{0-7}\}$. Examples of the applications of these functions can be found in Table 1. Following (Galley et al., 2006), we distinguish between *minimal* and *composed* rules. The composed rules are generated by combining a sequence of minimal rules.

3.2 Free attachment of target function words

3.2.1 Motivation

We explain the motivation for the present research using an example that was extracted from our training data, as shown in Figure 1. In the alignment of this example, three lines (in dot lines) are used to align *was* and *the* with *ga* (subject particle), and *was* with *ta* (past tense auxiliary verb). Under this alignment, we are forced to extract rules with relatively large tree fragments. For example, by applying the GHKM algorithm (Galley et al., 2004), a rule rooted at c0 will take c7, t4, c4, c19, t2, and c15 as the leaves. The final tree fragment, with a height of 7, contains 13 nodes. In order to ensure that this rule is used during decoding, we must generate subtrees with a height of 7 for c0. Suppose that the input forest is binarized and that $|E|$ is the average number of hyperedges of each node, then we must generate $O(|E|^{2^6-1})$ subtrees[4] for c0 in the worst case. Thus, the existence of these rules prevents the generalization ability of the final rule set that is extracted.

In order to address this problem, we tailor the alignment by ignoring these three alignment pairs in dot lines. For example, by ignoring the ambiguous alignments on the Japanese function words, we enlarge the frontier set to include from 12 to 19 of the 24 non-terminal nodes. Consequently, the number of extractable minimal rules increases from 12 (with three reordering rules rooted at c0, c1, and c2) to 19 (with five reordering rules rooted at c0, c1, c2, c5, and c17). With more nodes included in the frontier set, we can extract more minimal and composed monotonic/reordering rules and avoid extracting the less generalized rules with extremely large tree fragments.

3.2.2 Why chunking?

In the proposed algorithm, we use a target chunk set to constrain the *attachment explosion problem* because we use a packed parse forest instead of a 1-best tree, as in the case of (Galley et al., 2006). Multiple interpretations of unaligned function words for an aligned tree-string pair result in a derivation forest. Now, we have a packed parse forest in which each tree corresponds to a derivation forest. Thus, pruning free attachments of function words is practically important in order to extract composed rules from this "(derivation) forest of (parse) forest".

In the English-to-Japanese translation test case of the present study, the target chunk set is yielded by a state-of-the-art Japanese dependency parser, Cabocha v0.53[5] (Kudo and Matsumoto, 2002). The output of Cabocha is a list of *chunks*. A chunk contains roughly one content word (usually the head) and affixed function words, such as case markers (e.g., *ga*) and verbal morphemes (e.g., *sa re ta*, which indicate past tense and passive voice). For example, the Japanese sentence in Figure 1 is separated into four chunks, and the dependencies among these chunks are identified by arrows. These arrows point out the head chunk that the current chunk modifies. Moreover, we also hope to gain a fine-grained alignment among these syntactic chunks and source tree fragments. Thereby, during decoding, we are binding the generation of function words with the generation of target chunks.

[4] For one (binarized) hyperedge e of a node, suppose there are x subtrees in the left tail node and y subtrees in the right tail node. Then the number of subtrees guided by e is $(x+1) \times (y+1)$. Thus, the recursive formula is $N_h = |E|(N_{h-1}+1)^2$, where h is the height of the hypergraph and N_h is the number of subtrees. When $h = 1$, we let $N_h = 0$.

[5] http://chasen.org/∼taku/software/cabocha/

Algorithm 1 Aligning function words to the forest

Input: HPSG forest F_S, target sentence T, word alignment $A = \{(i, j)\}$, target function word set $\{f_w\}$ appeared in T, and target chunk set $\{\mathcal{C}\}$
Output: a derivation forest DF

$A' \leftarrow A \setminus \{(i, s(f_w))\}$ ▷ $f_w \in \{f_w\}$
for each node $v \in \mathcal{P}_{A'}$ in topological order **do**
  $\mathcal{T}_v \leftarrow \varnothing$ ▷ store the corresponding spans of v
  for each function word $f_w \in \{f_w\}$ **do**
    if $f_w \in \mathcal{C}$ and $t(v) \cap (\mathcal{C}) \neq \varnothing$ and f_w are not attached to descendants of v **then**
      append $t(v) \cup \{s(f_w)\}$ to $\mathcal{T}_v$
    end if
  end for
  for each corresponding span $t(v) \in \mathcal{T}_v$ **do**
    $\mathcal{R} \leftarrow$ IDENTIFYMINRULES$(v, t(v), T)$ ▷ range over the hyperedges of v, and discount the factional count of each rule $r \in \mathcal{R}$ by $1/|\mathcal{T}_v|$
    create a node n in DF for each rule $r \in \mathcal{R}$
    create a shared parent node $\oplus$ when $|\mathcal{R}| > 1$
  end for
end for

3.2.3 The algorithm

Algorithm 1 outlines the proposed approach to constructing a derivation forest to include multiple interpretations of target function words. The derivation forest is a hypergraph as previously used in (Galley et al., 2006), to maintain the constraint that one unaligned target word be attached to some node v exactly once in one derivation tree. Starting from a triple $\langle F_S, T, A\rangle$, we first tailor the alignment A to A' by removing the alignments for target function words. Then, we traverse the nodes $v \in \mathcal{P}_{A'}$ in topological order. During the traversal, a function word f_w will be attached to v if 1) $t(v)$ overlaps with the span of the chunk to which f_w belongs, and 2) f_w has not been attached to the descendants of v.

We identify translation rules that take v as the root of their tree fragments. Each tree fragment is a *frontier tree* that takes a node in the frontier set $\mathcal{P}_{\mathcal{A}'}$ of F_S as the root node and non-lexicalized frontier nodes or lexicalized non-frontier nodes as the leaves. Also, a *minimal frontier tree* used in a minimal rule is limited to be a frontier tree such that all nodes other than the root and leaves are non-frontier nodes. We use Algorithm 1 described in (Mi and Huang, 2008) to collect minimal frontier trees rooted at v in F_S. That is, we range over each hyperedges headed at v and continue to expand downward until the current set of hyperedges forms a minimal frontier tree.

	$A \rightarrow (A')$			
node	$s(\cdot)$	$t(\cdot)$	$c(\cdot)$	consistent
c0	0-6	0-8(0-3,5-7)	∅	1
c1	0-6	0-8(0-3,5-7)	∅	1
c2	0-6	0-8(0-3,5-7)	∅	1
c3	3-6	0-1,4-7(0-1, 5-7)	2,8	0
c4	3	5-7	0,8(0-3)	1
c5*	4-6	0,4(0-1)	2-8(2-3,5-7)	0(1)
c6*	0-3	2-8(2-3,5-7)	0,4(0-1)	0(1)
c7	0-1	2-3	0-1,4-8(0-1,5-7)	1
c8*	2-3	4-8(5-7)	0-4(0-3)	0(1)
c9	0	2	0-1,3-8(0-1,3,5-7)	1
c10	1	3	0-2,4-8(0-2,5-7)	1
c11	2-6	0-1,4-8(0-1,5-7)	2-3	0
c12	3	5-7	0,8(0-3)	1
c13*	5-6	0,4(0)	1-8(1-3,5-7)	0(1)
c14	5	4(∅)	0-8(0-3,5-7)	0
c15	6	0	1-8(1-3,5-7)	1
c16	2	4,8(∅)	0-7(0-3,5-7)	0
c17*	4-6	0,4(0-1)	2-8(2-3,5-7)	0(1)
c18	4	1	0,2-8(0,2-3,5-7)	1
c19	4	1	0,2-8(0,2-3,5-7)	1
c20*	0-3	2-8(2-3,5-7)	0,4(0-1)	0(1)
c21	3	5-7	0,8(0-3)	1
c22	2	4,8(∅)	0-7(0-3,5-7)	0
c23*	2-3	4-8(5-7)	0-4(0-3)	0(1)

Table 1: Change of node attributes after alignment modification from A to A' of the example in Figure 1. Nodes with * superscripts are consistent with A' but not consistent with A.

In the derivation forest, we use $\oplus$ nodes to manage minimal/composed rules that share the same node and the same corresponding span. Figure 2 shows some minimal rule and $\oplus$ nodes derived from the example in Figure 1.

Even though we bind function words to their nearby chunks, these function words may still be attached to relative large tree fragments, so that richer syntactic information can be used to predict the function words. For example, in Figure 2, the tree fragments rooted at node c_0^{0-8} can predict *ga* and/or *ta*. The syntactic foundation behind is that, whether to use *ga* as a subject particle or to use *wo* as an object particle depends on both the left-hand-side noun phrase (*kekka*) and the right-hand-side verb (*kensyou sa re ta*). This type of node v' (such as c_0^{0-8}) should satisfy the following two heuristic conditions:

- v' is included in the frontier set $\mathcal{P}_{\mathcal{A}'}$ of F_S, and
- $t(v')$ covers the function word, or v' is the root node of F_S if the function word is the beginning or ending word in the target sentence T.

Starting from this derivation forest with minimal

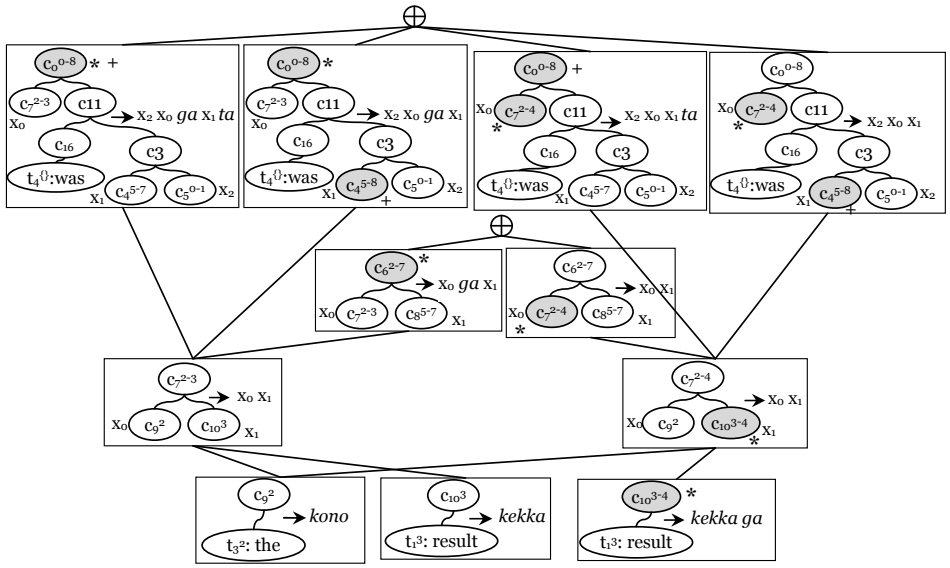

Figure 2: Illustration of a (partial) derivation forest. Gray nodes include some unaligned target function word(s). Nodes annotated by "*" include *ga*, and nodes annotated by "+" include *ta*.

rules as nodes, we can further combine two or more minimal rules to form composed rules nodes and can append these nodes to the derivation forest.

3.3 Estimating rule probabilities

We use the EM algorithm to jointly estimate 1) the translation probabilities and fractional counts of rules and 2) the scores of derivations in the derivation forests. As reported in (May and Knight, 2007), EM, as has been used in (Galley et al., 2006) to estimate rule probabilities in derivation forests, is an iterative procedure and prefers shorter derivations containing large rules over longer derivations containing small rules. In order to overcome this bias problem, we discount the fractional count of a rule by the product of the probabilities of parse hyperedges that are included in the tree fragment of the rule.

4 Experiments

4.1 Setup

We implemented the forest-to-string decoder described in (Mi et al., 2008) that makes use of forest-based translation rules (Mi and Huang, 2008) as the baseline system for translating English HPSG forests into Japanese sentences. We analyzed the performance of the proposed translation rule sets by using the same decoder.

	Train	Dev.	Test
# sentence pairs	994K	2K	2K
# En 1-best trees	987,401	1,982	1,984
# En forests	984,731	1,979	1,983
# En words	24.7M	50.3K	49.9K
# Jp words	28.2M	57.4K	57.1K
# Jp function words	8.0M	16.1K	16.1K

Table 2: Statistics of the JST corpus. Here, En = English and Jp = Japanese.

The JST Japanese-English paper abstract corpus[6] (Utiyama and Isahara, 2007), which consists of one million parallel sentences, was used for training, tuning, and testing. Table 2 shows the statistics of this corpus. Note that Japanese function words occupy more than a quarter of the Japanese words. Making use of Enju 2.3.1, we generated 987,401 1-best trees and 984,731 parse forests for the English sentences in the training set, with successful parse rates of 99.3% and 99.1%, respectively. Using the pruning criteria expressed in (Mi and Huang, 2008), we continue to prune a parse forest by setting p_e to be 8, 5, and 2, until there are no more than $e^{10} = 22,026$ trees in a forest. After pruning, there are an average of 82.3 trees in a parse forest.

[6] http://www.jst.go.jp

	C3-T	M&H-F	Min-F	C3-F
free fw	Y	N	Y	Y
alignment	A'	A	A'	A'
English side	tree	forest	forest	forest
# rule	86.30	96.52	144.91	228.59
# reorder rule	58.50	91.36	92.98	162.71
# tree types	21.62	93.55	72.98	120.08
# nodes/tree	14.2	42.1	26.3	18.6
extract time	30.2	52.2	58.6	130.7
EM time	9.4	-	11.2	29.0
# rules in dev.	0.77	1.22	1.37	2.18
# rules in test	0.77	1.23	1.37	2.15
DT(sec./sent.)	2.8	15.7	22.4	35.4
BLEU (%)	26.15	27.07	27.93	**28.89**

Table 3: Statistics and translation results for four types of tree-to-string rules. With the exception of '# nodes/tree', the numbers in the table are in millions and the time is in hours. Here, fw denotes function word, and DT denotes the decoding time, and the BLEU scores were computed on the test set.

We performed GIZA++ (Och and Ney, 2003) and the *grow-diag-final-and* symmetrizing strategy (Koehn et al., 2007) on the training set to obtain alignments. The SRI Language Modeling Toolkit (Stolcke, 2002) was employed to train a five-gram Japanese LM on the training set. We evaluated the translation quality using the BLEU-4 metric (Papineni et al., 2002).

Joshua v1.3 (Li et al., 2009), which is a freely available decoder for hierarchical phrase-based SMT (Chiang, 2005), is used as an external baseline system for comparison. We extracted 4.5M translation rules from the training set for the 4K English sentences in the development and test sets. We used the default configuration of Joshua, with the exception of the maximum number of items/rules, and the value of k (of the k-best outputs) is set to be 200.

4.2 Results

Table 3 lists the statistics of the following translation rule sets:

- C3-T: a composed rule set extracted from the derivation forests of 1-best HPSG trees that were constructed using the approach described in (Galley et al., 2006). The maximum number of internal nodes is set to be three when generating a composed rule. We free attach target function words to derivation forests;

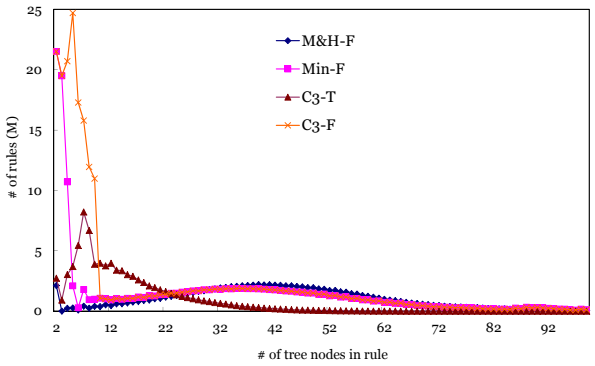

Figure 3: Distributions of the number of tree nodes in the translation rule sets. Note that the curves of Min-F and C3-F are duplicated when the number of tree nodes being larger than 9.

- M&H-F: a minimal rule set extracted from HPSG forests using the extracting algorithm of (Mi and Huang, 2008). Here, we make use of the original alignments. We use the two heuristic conditions described in Section 3.2.3 to attach unaligned words to some node(s) in the forest;

- Min-F: a minimal rule set extracted from the derivation forests of HPSG forests that were constructed using Algorithm 1 (Section 3).

- C3-F: a composed rule set extracted from the derivation forests of HPSG forests. Similar to C3-T, the maximum number of internal nodes during combination is three.

We investigate the generalization ability of these rule sets through the following aspects:

1. the number of rules, the number of reordering rules, and the distributions of the number of tree nodes (Figure 3), i.e., more rules with relatively small tree fragments are preferred;

2. the number of rules that are applicable to the development and test sets (Table 3); and

3. the final translation accuracies.

Table 3 and Figure 3 reflect that the generalization abilities of these four rule sets increase in the order of C3-T $<$ M&H-F $<$ Min-F $<$ C3-F. The advantage of using a packed forest for re-alignment is verified by comparing the statistics of the rules and

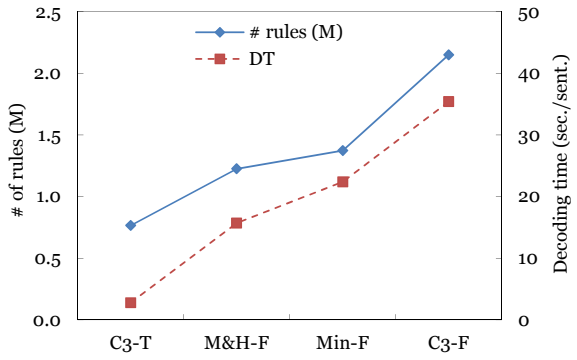

Figure 4: Comparison of decoding time and the number of rules used for translating the test set.

the final BLEU scores of C3-T with Min-F and C3-F. Using the composed rule set C3-F in our forest-based decoder, we achieved an optimal BLEU score of 28.89 (%). Taking M&H-F as the baseline translation rule set, we achieved a significant improvement ($p < 0.01$) of 1.81 points.

In terms of decoding time, even though we used Algorithm 3 described in (Huang and Chiang, 2005), which lazily generated the N-best translation candidates, the decoding time tended to be increased because more rules were available during cube-pruning. Figure 4 shows a comparison of decoding time (seconds per sentence) and the number of rules used for translating the test set. Easy to observe that, decoding time increases in a nearly linear way following the increase of the number of rules used during decoding.

Finally, compared with Joshua, which achieved a BLEU score of 24.79 (%) on the test set with a decoding speed of 8.8 seconds per sentence, our forest-based decoder achieved a significantly better ($p < 0.01$) BLEU score by using either of the four types of translation rules.

5 Related Research

Galley et al. (2006) first used derivation forests of aligned tree-string pairs to express multiple interpretations of unaligned target words. The EM algorithm was used to jointly estimate 1) the translation probabilities and fractional counts of rules and 2) the scores of derivations in the derivation forests. By dealing with the ambiguous word alignment instead of unaligned target words, syntax-based re-alignment models were proposed by (May and Knight, 2007; Wang et al., 2010) for tree-based translations.

Free attachment of the unaligned target word problem was ignored in (Mi and Huang, 2008), which was the first study on extracting tree-to-string rules from aligned forest-string pairs. This inspired the idea to re-align a packed forest and a target sentence. Specially, we observed that most incorrect or ambiguous word alignments are caused by function words rather than content words. Thus, we focus on the realignment of target function words to source tree fragments and use a dependency parser to limit the attachments of unaligned target words.

6 Conclusion

We have proposed an effective use of target function words for extracting generalized transducer rules for forest-based translation. We extend the unaligned word approach described in (Galley et al., 2006) from the 1-best tree to the packed parse forest. A simple yet effective modification is that, during rule extraction, we account for multiple interpretations of both aligned and unaligned target function words. That is, we chose to loose the ambiguous alignments for all of the target function words. The consideration behind is in order to generate target function words in a robust manner. In order to avoid generating too large a derivation forest for a packed forest, we further used chunk-level information yielded by a target dependency parser. Extensive experiments on large-scale English-to-Japanese translation resulted in a significant improvement in BLEU score of 1.8 points ($p < 0.01$), as compared with our implementation of a strong forest-to-string baseline system (Mi et al., 2008; Mi and Huang, 2008).

The present work only re-aligns target function words to source tree fragments. It will be valuable to investigate the feasibility to re-align all the target words to source tree fragments. Also, it is interesting to automatically learn a word set for re-aligning[7]. Given source parse forests and a target word set for re-aligning beforehand, we argue our approach is generic and applicable to any language pairs. Finally, we intend to extend the proposed approach to tree-to-tree translation frameworks by

[7]This idea comes from one reviewer, we express our thankfulness here.

re-aligning subtree pairs (Liu et al., 2009; Chiang, 2010) and consistency-to-dependency frameworks by re-aligning consistency-tree-to-dependency-tree pairs (Mi and Liu, 2010) in order to tackle the rule-sparseness problem.

Acknowledgments

The present study was supported in part by a Grant-in-Aid for Specially Promoted Research (MEXT, Japan), by the Japanese/Chinese Machine Translation Project through Special Coordination Funds for Promoting Science and Technology (MEXT, Japan), and by Microsoft Research Asia Machine Translation Theme.

Wu (wu.xianchao@lab.ntt.co.jp) has moved to NTT Communication Science Laboratories and Tsujii (junichi.tsujii@live.com) has moved to Microsoft Research Asia.

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270, Ann Arbor, MI.

David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452, Uppsala, Sweden, July. Association for Computational Linguistics.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of HLT-NAACL*.

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 COLING-ACL*, pages 961–968, Sydney.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of IWPT*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edomonton, Canada, May 27-June 1.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 177–180.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of CoNLL-2002*, pages 63–69. Taipei, Taiwan.

Zhifei Li, Chris Callison-Burch, Chris Dyery, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Jonathan Weese, and Omar F. Zaidan. 2009. Demonstration of joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 25–28, August.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment templates for statistical machine transaltion. In *Proceedings of COLING-ACL*, pages 609–616, Sydney, Australia.

Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. In *Proceedings of ACL-IJCNLP*, pages 558–566, August.

Samuel E. Martin. 1975. *A Reference Grammar of Japanese*. New Haven, Conn.: Yale University Press.

Jonathan May and Kevin Knight. 2007. Syntactic re-alignment models for machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 360–368, Prague, Czech Republic, June. Association for Computational Linguistics.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of EMNLP*, pages 206–214, October.

Haitao Mi and Qun Liu. 2010. Constituency to dependency translation with forests. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1433–1442, Uppsala, Sweden, July. Association for Computational Linguistics.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of ACL-08:HLT*, pages 192–199, Columbus, Ohio.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic hpsg parsing. *Computational Lingustics*, 34(1):35–80.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. Number 152 in CSLI Lecture Notes. CSLI Publications.

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

Masao Utiyama and Hitoshi Isahara. 2007. A japanese-english patent parallel corpus. In *Proceedings of MT Summit XI*, pages 475–482, Copenhagen.

Wei Wang, Jonathan May, Kevin Knight, and Daniel Marcu. 2010. Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. *Computational Linguistics*, 36(2):247–277.

Xianchao Wu, Takuya Matsuzaki, and Jun'ichi Tsujii. 2010. Fine-grained tree-to-string translation rule extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 325–334, Uppsala, Sweden, July. Association for Computational Linguistics.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of HLT-NAACL*, pages 245–253.

Min Zhang, Hongfei Jiang, Ai Ti Aw, Jun Sun, Sheng Li, and Chew Lim Tan. 2007. A tree-to-tree alignment-based model for statistical machine translation. In *Proceedings of MT Summit XI*, pages 535–542, Copenhagen, Denmark, September.