A Synchronous Context Free Grammar using Dependency Sequence for Syntax-based Statistical Machine Translation

Hwidong Na **Jin-Ji Li** **Yeha Lee** **Jong-Hyeok Lee**
Division of Electrical and Computer Engineering,
Pohang University of Science and Technology (POSTECH),
San 31 Hyoja Dong, Pohang, 790-784, Republic of Korea
{leona,ljj,sion,jhlee}@postech.ac.kr

Abstract

We introduce a novel translation rule that captures discontinuous, partial constituent, and non-projective phrases from source language. Using the traversal order sequences of the dependency tree, our proposed method 1) extracts the synchronous rules in linear time and 2) combines them efficiently using the CYK chart parsing algorithm. We analytically show the effectiveness of this translation rule in translating relatively free order sentences, and empirically investigate the coverage of our proposed method.

1 Introduction

Statistical machine translation (SMT) has been the dominant research area of machine translation. SMT frameworks usually extract translation rules, i.e. a pair of the source and target unit, automatically from a parallel corpus. At the extraction stage, phrase-based SMT frameworks regard a pair of continuous word sequences, i.e. phrases, in the source and the target sentence as a translation rule. One of the major drawbacks of phrase-based frameworks is that reordering of phrases to generate a grammatical (target) sentence is difficult without syntactic information.

Most of the recent work has developed a SMT model that integrated syntactic information such as a tree structure based on context free grammar (CFG) or dependency grammar (DG), refered to as syntax-based SMT frameworks. They define translation rules that encode global reordering information on the source and/or the target side. Using placeholders (variables) for other translation units, translation rules in syntax-based SMT frameworks embed hierarchical properties. In other words, they are capable of translating discontinuous phrases, while conventional phrase-based SMT systems are unable [1]. A discontinuous source phrase could be translated into a (continuous) target phrase, and vice versa. For example, “who . . . she” is translated into a continuous target phrase in Figure 1.

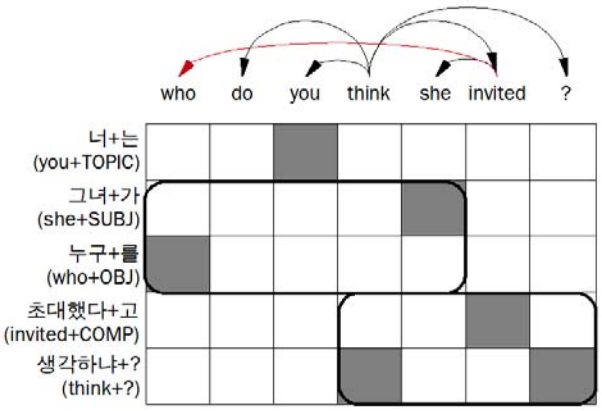


Figure 1: A word-aligned sentence pair with a non-projective dependency from “invited” to “who”. A source phrase “who . . . invited” is discontinuous and partial constituent phrase, where its corresponding target phrase is continuous.

A major challenge of syntax-based SMT frameworks is to broaden the coverage of translation rules. Purely syntactic translation rules allow only

[1] Galley and Manning (2010) proposed a phrase-based SMT system which supports discontinuous phrases

constituent phrases as translation units (Galley et al., 2004). This restriction is too severe to capture frequent patterns that are smaller than constituent (partial constituent) phrases. For example, an English phrase like “something CC” can be translated into a Japanese phrase “(something) (CC)” where the parentheses mean their translated counterparts. Modifiers in a noun phrase are also a partial constituent phrase as a translation unit. In Figure 1, “who ... she” is a partial constituent phrase. Many researchers have integrated partial constituent phrases into translation units.

Since a tree structure would require a non-projective relation, supporting non-projective dependency helps broaden the coverage of translation rules. For example, a dependency relation from “invited” from “who” is non-projective in Figure 1. Formally, a non-projective dependency is a relation from a head w_i to a dependent w_j such that $\exists head(w_k) \notin [min(i,j), max(i,j)]$ where $k \in [min(i,j), max(i,j)]$ (i, j, and k are indices). DG handles non-projective relations much more easily than CFG, and is also known to be more suitable at handling divergences between two languages than the other formalisms (Fox, 2002).

Last but not least, a source sentence would have a relatively free order. Languages with relatively free order such as Korean and Japanese allow various types of ordering of dependents for a given head. Especially for the main predicate of a source sentence, the modifiers such as the subjects and the objects can be located any position before the main predicate. For example, the following six Korean sentences have different word orders but identical meaning “when does the train leave to Seoul?” in English.

Modifiers in flexible word order			Head
기차+가	서울+로	언제	출발합니까+?
서울+로	기차+가	언제	
언제	서울+로	기차+가	
언제	기차+가	서울+로	
서울+로	언제	기차+가	
기차+가	언제	서울+로	

where the main predicate “출발합니까+?(leave+?)” shared for each sentence has three modifiers “기차+가(train+SUBJ)”, “서울+로(Seoul+to)”, and “언제(when)”. In order to broaden the coverage of translation rules, handling the relatively free order of the source sentence would be useful in this case.

We introduce a novel translation rule to manipluate discontinuous, partial constituent, and non-projective phrases using the dependency tree of the source language. Our proposed method also allows that the source sentence has relatively free order. The key idea is to traverse the dependency tree and regard the sequences of the traversal order as phrases (Section 3). We define a bilingual synchronous grammar, which can simultaneously generate the sequence in the source language and the target sentence (Section 4). The rule extraction algorithm runs in linear time by restricting the sequences (Section 5). The extracted rules are combined efficiently using a CYK chart parsing algorithm (Section 6). We analytically show the effectiveness (Section 7), and empirically investigate the coverage of our proposed method (Section 8).

2 Related Work

It is presumably intractable to extract discontinuous phrases exhaustively. Rather we need a restricted method that leverages the coverage and the computational efficiency of the extraction. Since words in head-modifier relations are more colsely related than the others in a sentence, many syntax-based SMT systems use DG based on dependency treelets, which are connected subgraphes. A dependency treelet would be discontinuous and therefore useful to extract discontinuous phrases. For example, “who ... she invited” in Figure 1 is a dependency treelet and discontinuous in the source sentence. Some approaches using dependency treelets assumed the isomorphism of the dependency structure of the source and the target sentence (Lin, 2004; Quirk et al., 2005), which is unrealistic in the real situation.

Although other approaches using dependency treelets addressed the non-isomorphism (Eisner, 2003; Ding and Palmer, 2005; Xiong et al., 2007), dependency treelets cannot capture partial constituent phrases such as sequences of dependents. It would cause the low coverage of translation rules since modifiers under a common head are often

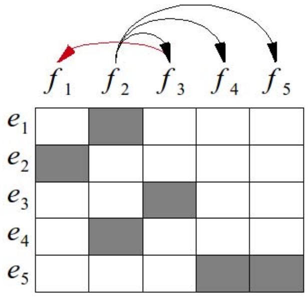


Figure 2: An example of a pair of the source sentence with the dependency tree $H = 30222$ and the target sentence. The postorder is $P = 15234$ and the breadth first order is $B = 51234$. Therefore, $PS_1^5 = (1, 3, 4, 5, 2)$ and $BS_1^5 = (2, 3, 4, 5, 1)$ by definition.

Table 1: The PSs of the example when we restrict the maximum length of PS to 2. An underline means that the PS conflicts.

sequence	type	aspan	cspan
$PS_1^1 = (1)$	treelet	[2,2]	[2,2]
$PS_2^2 = (3)$	treelet	[3,3]	[2,3]
$\underline{PS_3^3 = (4)}$	treelet	[5,5]	[5,5]
$\underline{PS_4^4 = (5)}$	treelet	[5,5]	[5,5]
$PS_5^5 = (2)$	treelet	[1,4]	[1,5]
$PS_1^2 = (1, 3)$	treelet	[2,3]	[2,3]
$\underline{PS_2^3 = (3, 4)}$	treeseq	[3,5]	[2,5]
$PS_3^4 = (4, 5)$	treeseq	[5,5]	[5,5]
$\underline{PS_4^5 = (5, 2)}$	treelet	[1,5]	[1,5]

translated as patterns. Shen et al. (2008) introduced tree sequences, i.e. sequences of treelets, as well as treelets in a dependency tree. They reduce the search space for decoding by restricting the extracted translation units of the dependency structure on the target language. Nevertheless, they were not able to handle non-projective dependencies.

Carreras and Collins (2009) attempted to allow arbitrary reordering of the source language using tree adjoining grammar. None of the previous work using DG, however, incorporated the relatively free order in the source language as our proposed method.

3 Dependency Sequence

Our key observation is that a graph traversal of a dependency tree leads to the discovery of useful patterns. The patterns in the form of treelets and tree sequences would be discontinuous and partial constituent phrases. For example, by the postorder traversal we visit English words in Figure 1 in the following order: "do, you, who, she, invited, ?, think". Among the sequence of words by the postorder traversal, a subsequence would be a discontinuous treelet ("think . . . ?") or a tree sequence of a patial constituent ("who . . . she"). The subsequences would have non-projective dependency as well. We refer the subsequences by a graph traversal as the *depedenecy sequences*.

Let us define the dependency seqeunce formally. For a source sentence $F = f_1 \cdots f_n$, let $H = h_1 \cdots h_n$ be the dependency tree of F where the head of f_j is f_{h_j}. Dependency sequences, for instance, would be discontinuous ($f_1 \ \ldots f_3$) or partial constituent ($f_4 f_5$) under common head (f_2). Let $P = p_1 \cdots p_n$ denote the visiting sequence by the postorder traversal of H, and $B = b_1 \cdots b_n$ denote the analogy of the breadth first order. P and B is defined even if the dependency tree has non-projective dependencies. Note that $p_j = j$ if the source language is a head final language such as Korean or Japanese. Figure 2 shows an example of F , H, P and B in a dependency tree with a non-projective dependency ($h_1 = 3$).

Let $PS_m^l = (j_m, \cdots, j_l)$ be a dependency sequence in the postorder in the dependency tree, where $p_{j_m} < \cdots < p_{j_l}$. We first define a dependent PS_p^q of PS_m^l as follows:

Definition 1. *A PS_p^q is a **dependent** of PS_m^l if $\exists j_k \in PS_p^q, h_{j_k} \in PS_m^l$ and $q < m$.*

There are two types of PSs: treelets and treeseqs.

Definition 2. *A **treelet** PS_m^l is a connected subgraph of the dependency tree. The root of the treelet is at the end, i.e. $\forall j_k$ s.t. $p_{j_k} < p_{j_l}$, $h_{j_k} \in PS_m^l$ or $h_{j_k} = j_l$.*

Definition 3. *A **treeseq** PS_m^l is an ordered set of treelets which are dependents of a common head f_{j_h}, i.e. $\forall j_k$ s.t. $p_{j_m} \leq p_{j_k} \leq p_{j_l}$, $h_{j_k} \in PS_m^l$ or $h_{j_k} = j_h$ $(p_{j_l} < p_{j_h})$.*

On the other hand, let $BS_m^l = (j_m, \cdots, j_l)$ be a dependency sequence in the breadth first order in the dependency tree, where $b_{j_m} < \cdots < b_{j_l}$. There are also two types of BSs.

Definition 4. *A **treelet** BS_m^l is a connected subgraph of the dependency tree. The root of the treelet* *is at the beginning, i.e.* $\forall j_k$ *s.t.* $b_{j_m} < b_{j_k}$, $h_{j_k} \in BS$ *or* $h_{j_k} = j_m$

Definition 5. *A* ***treeseq*** BS_m^l *is an ordered set of treelets which are dependents of the common head* f_{j_h}*, i.e.* $\forall j_k$ *s.t.* $b_{j_m} \leq b_{j_k} \leq b_{j_l}$, $h_{j_k} \in BS_m^l$ *or* $h_{j_k} = j_h$ $(b_{j_h} < b_{j_m})$*.*

Figure 2 shows that $PS_1^2 = (1, 3)$ is a discontinuous and non-projective phrase and $PS_3^4 = (4, 5)$ is a partial constituent phrase. Therefore, a PS (we omit the index m, l for brevity) could be a discontinuous and partial constituent translation unit in the source language. Note that they also allow non-projectivity because a PS is defined regardless of projectivity. Table 1 shows the PSs of Figure 2 where the type is either a treelet or a treeseq as defined above. The BSs are omitted for brevity.

The number of possible PSs or BSs for the source sentence consisting of n words is $\frac{n^2+n}{2}$ at most. In order to identify PSs and BSs more efficiently, we introduce alignment information to PS (or BS), defined as follows:

Definition 6. *An* ***aligned span*** *of* PS*, denoted by* $aspan(PS)$*, is the word sequence in a target sentence ranging from the lower bound to the upper bound according to the set of word alignments.*

Lin (2004) and Xiong et al. (2007) used similar notation to the aligned span, calling it "head span" and "word span", respectively. They also defined the *union* of the aligned span rooted at the given node as a "phrase span" and a "node span", respectively. Conceptually, the same definition is used for each PS which is a sequence of nodes.

Definition 7. *A* ***covered span*** *of* PS*, denoted by* $cspan(PS)$*, is the word sequence in a target sentence that ranges from the lower bound to the upper bound of the aligned set for all nodes in subtrees that have their root in the* PS *as well as the* PS *itself.*

Note that $apan(PS) \subseteq cspan(PS)$ and $cspan(PS)$ is identified in linear time according to the postorder.

4 Synchronous Context Free Grammar (SCFG) using Dependency Sequence

We propose a novel grammar approach using PS (or BS) in the SCFG framework. This incorporates the merits of both PS and SCFG. At the same time as we retain its non-isomorphic construction capability, we capture discontinuous, partial constituent, non-projective phrases in the source sentence. Intuitively, a PS is a surrogate of non-terminals in CFG, which is replaced with other non-terminals or terminal symbols. We give a formal and general definition of a synchronous grammar using PS as follows:

Table 2: All possible extracted production rules for the example. An underline means that the rule is not minimal.

Rule:	$\langle \quad \alpha \ , \ \beta \ \rangle$
γ_1:	$\langle\, PS_1^1 \ , \ e_2 \ \rangle$
γ_2:	$\langle\, PS_2^2 \ , \ PS_1^1 \ e_3 \ \rangle$
$\underline{\gamma_3}$:	$\langle\, PS_1^2 \ , \ e_2 e_3 \ \rangle$
γ_4:	$\langle\, PS_3^4 \ , \ e_5 \ \rangle$
γ_5:	$\langle\, PS_5^5 \ , \ e_1 \ PS_2^2 \ e_4 \ PS_3^4 \ \rangle$
$\underline{\gamma_6}$:	$\langle\, PS_5^5 \ , \ e_1 \ PS_1^2 \ e_4 \ PS_3^4 \ \rangle$

Definition 8. *A SCFG using PS* ***(SCFG-PS)*** *is a 5-tuple* $G = \langle \Sigma_S, \Sigma_T, \Delta, \Gamma, \Theta \rangle$*, where:*

- Σ_S and Σ_T are finite sets of terminals (words, POSs, etc.) of the source and target languages, respectively.

- Δ is a finite set of PSs in the source language.

- Γ is a finite set of production rules where a production rule $\gamma : X \rightarrow \langle\ \alpha\ ,\ \beta\ \rangle$, which is a relationship from Δ to $\{\Delta \cup \Sigma_T\} *$. The asterisk represents the Kleenstar operation.

- Θ is the start symbol used to represent the whole sentence, i.e. $\gamma_0 : \Theta \rightarrow \langle\ X\ ,\ X\ \rangle$

The definition for BS is omitted because it is identical to the PS case.

Note that the β of a production rule contains PS regardless of its position in the dependency tree of the source sentence. In other words, we can handle the relative free order in the source language during the synchronous derivation. We will explain this in Section 7.

5 Rule Extraction

In this section, we illustrate the extraction algorithm for SCFG-PS. Because we regard PSs as non-terminals, it is the α of a production rule γ. If PS appers in β, it means that the PS is replaced with

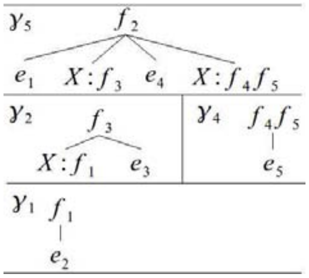


Figure 3: A visual representation of the minimal rules in Table 2

β' of PS if $\langle\ PS\ ,\ \beta'\ \rangle\ \in\ \Gamma$. This substitution is called the derivation. At the end of the derivation, β is a sequence of the target words. Therefore, we extract production rules which make the derivation possible.

Let PS^q_p be a dependent of PS^l_m. We extract a production rule where $PS^l_m \in \alpha$ and $PS^q_p \in \beta$. Because $cspan(PS^q_p) \subseteq cspan(PS^l_m)$, β includes the target words in $cspan(PS^l_m)$, but excludes them in $cspan(PS^q_p)$. We allow the extraction only if the covered span of PS^q_p is a subset of the aligned span of the PS^l_m, or a disjoint span. In other words, we do not extract useless production rules, which cannot derive the target sentence. In Figure 2, the dependent $PS^2_2 = (1,3)$ of $PS^5_5 = (2)$ is allowed, but the dependent $PS^3_2 = (3,4)$ is not.

Formally, we extract non-*conflict* PSs defined as follows.

Definition 9. *A PS^l_m is **consistent** with PS^q_p if it satisfies one of the following conditions:*

- $cspan(PS^q_p) \subset aspan(PS^l_m)$
- $cspan(PS^q_p) \cap aspan(PS^l_m) = \emptyset$

Definition 10. *A PS^l_m **conflicts** if it satisfies one of the following conditions:*

- $aspan(PS^l_m) = \emptyset$
- $\forall PS^q_p$, PS^l_m is not *consistent* with PS^q_p, where PS^q_p is a dependent of PS^l_m
- $\forall PS^s_r, PS^s_r$ is not *consistent* with PS^l_m, where PS^l_m is a dependent of PS^s_r
- $\exists PS^u_t$ s.t. $cspan(PS^u_t) \cap cspan(PS^l_m) \neq \emptyset$, where $\forall k \in [t,u], k \notin [p,q]$ and $k \notin [m,l]$ and $k \notin [r,s]$.

Algorithm 1 Extract

```
1: Input: the sequence of the source sentence
   $(j_1 \dots j_n)$ where $p_{j_k} < p_{j_{k+1}} \forall k \in [1, n-1]$
2: for each minimal $PS^l_m$ do
3:     $\beta \leftarrow$ target words in $cspan(PS^l_m)$
4:     for each minimal dependent $PS^q_p$ do
5:         $\beta \leftarrow$ substitute target words in
           $cspan(PS^q_p)$ for $PS^q_p$ from $\beta$
6:     end for
7:     yield a production rule $\gamma$ : $\langle\ PS^l_m\ ,\ \beta\ \rangle$
8: end for
```

Although we only extract non-conflicting PS, the enumeration takes a $O(n^2)$ time. It can be reduced, however, if we extract only the *minimal* PSs defined as follows:

Definition 11. *A PS^l_m is **separable** if it satisfies the following conditions:*

- $m < l$, and
- $\exists k \in [m+1, l]$, both $PS^{j_{k-1}}_{j_m}$ and $PS^{j_l}_{j_k}$ do not conflict

Definition 12. *A PS is **minimal** if it satisfies the following conditions:*

- The PS does not conflict, and
- The PS is not separable

The definitions for BS are analogous.

Algorithm 1 shows that we extract the production rules γ for each minimal PS^l_m (line 2). We also introduce the restriction to β that the substituted sequence PS^q_p are minimal (line 4). Therefore, PSs in β are also minimal in γ. Table 2 shows that γ_3 and γ_6 are not minimal rules because $PS^2_1 = (1,3)$ is seperable into $PS^1_1 = (1)$ and $PS^2_2 = (3)$. The complexity becomes $O(n)$ because we have a disjoint set of PS, i.e. $\forall PS^{l_k}_{m_k} \in \Delta$, $\neg\exists PS^l_m$ s.t. $m_k \leq m \leq l_k$ or $m \leq l_k \leq l$.

6 Rule Combinination

The extracted rules are Γ of SCFG-PS as defined in Section 4. The combinaions of rules can be regarded as a series of synchronous derivation steps from the start symbol Θ. For instance, the PS^5_1 and the target sentence in Figure 2 is generated as follows:

Algorithm 2 Combine

```
Input: the extracted rules $\Gamma$, and
the sequence of the source sentence $(j_1 \ldots j_n)$
initialize chart $C$ with $\Gamma$
for each $m = 1$ to $n$ do
    for each $l = m + 1$ to $n$ do
        for each $k = m$ to $l$ do
            for each $\langle PS_m^l\ ,\ \beta\rangle \in \Gamma$ do
                if $PS_m^k \in \beta$
                and $PS_{k+1}^l \in \beta$ then
                    store $PS_m^l$ to $C$
                end if
            end for
        end for
    end for
end for
if $PS_1^n \in C$ then
    generate the target string
end if
```

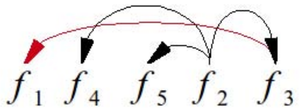


Figure 4: A sentence with the same words and dependency relations but different orderings, where $PS_1^5 = (4, 5, 1, 3, 2)$

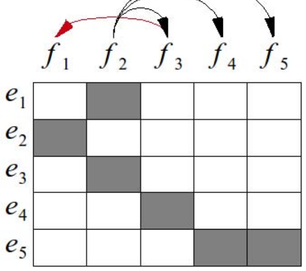


Figure 5: A degenerate case for the proposed method using $PS_1^5 = (1, 3, 4, 5, 2)$. Note that $BS_1^5 = (2, 3, 4, 5, 1)$ works as a complementarity.

$$\begin{array}{lll} & & \langle\ X\ ,\ X\ \rangle \\ \text{by } \gamma_5\text{:} & \Rightarrow & \langle\ PS_5^5\ \ ,\ e_1\ PS_2^2\ e_4\ PS_3^4\ \rangle \\ \text{by } \gamma_4\text{:} & \Rightarrow & \langle\ PS_3^5\ \ ,\ e_1\ PS_2^2\ e_4\ [\ e_5\ ]\ \rangle \\ \text{by } \gamma_2\text{:} & \Rightarrow & \langle\ PS_2^5\ \ ,\ e_1\ [\ PS_1^1\ e_3\ ]\ e_4\ [e_5]\ \rangle \\ \text{by } \gamma_1\text{:} & \Rightarrow & \langle\ PS_1^5\ \ ,\ e_1\ [\ [\ e_2\ ]\ e_3\ ]\ e_4\ [e_5]\ \rangle \end{array}$$

where the bracket is used to represent each step of production, for convenience.

To combine the rules, we adopt the CYK chart parsing algorithm, which regards the span $[m, l]$ of the chart as PS_m^l. A PS_m^l combines two subsequences PS_m^k and PS_{k+1}^l values which are stored in the chart as shown in Algorithm 2.

7 Analysis

7.1 A relatively free order

The proposed method makes it possible to translate relatively free order sentences in the source language. Figure 4 shows another example. The example has different orders of traversal PS_1^5, while words and dependency relations are identical to the source sentence in Figure 2. Nevertheless, the target sentence can be generated because we do not restrict the relative order of the dependents:

$$\begin{array}{lll} & & \langle\ X\ ,\ X\ \rangle \\ \text{by } \gamma_5\text{:} & \Rightarrow & \langle\ PS_5^5\ \ ,\ e_1\ PS_2^2\ e_4\ PS_3^4\ \rangle \\ \text{by } \gamma_2\text{:} & \Rightarrow & \langle\ PS_4^5\ \ ,\ e_1\ [\ PS_1^1\ e_3\ ]\ e_4\ PS_3^4\ \rangle \\ \text{by } \gamma_1\text{:} & \Rightarrow & \langle\ PS_3^5\ \ ,\ e_1\ [\ [\ e_2\ ]\ e_3\ ]\ ]\ e_4\ PS_3^4\ \rangle \\ \text{by } \gamma_4\text{:} & \Rightarrow & \langle\ PS_1^5\ \ ,\ e_1\ [\ [\ e_2\ ]\ e_3\ ]\ e_4\ [\ e_5\ ]\ \rangle \end{array}$$

For example, the subject can preceed the object of the main predicate, or vice versa in Korean. In this case, a translation rule specifying the order of the subject and the object fails to capture the relatively free order. However, our method is applicable to both structure without any special treatment.

7.2 Complementarity: PS and BS

There is a weakness in the proposed method using PS because it cannot deal with non-projective target phrases. Figure 5 shows a degenerate case. If we assume that the dependencies in the target sentence are obtained by projecting the source dependencies, the target sentence in Figure 5 has a non-projective dependency $h_1^e = 3$ where $h_k^e = i$ denotes that e_i is the head of e_k.

BS is an alternative for this reason because it defines a different order from PS. The minimal sequence BS_1^2 in Figure 5 captures the non-projective target phrase using the production rule $\gamma\ :\ \langle\ BS_1^2,\ \ e_1\ BS_5^5\ e_3\ BS_3^4\ \rangle$. Therefore, we expect PS and BS to be complementary.

7.3 Non-projective dependency

Trees with non-projective dependencies appear quite often in some languages such as Czech and Danish (Nivre, 2006). Recent work on dependency parsing has suggested various methods for non-projective dependencies. The proposed method easily deals with non-projective source phrases because the sequence PS is always defined in the dependency tree. Table 2 shows an extracted rule γ_3 where the $PS_2^2 = (3)$ has the dependents $PS_1^1 = (1)$ and the relation between them is non-projective.

8 Emperical Result and Discussion

8.1 Experiment and environment

To investigate the coverage of the extracted translation rules, we extracted the rules from the training corpus and re-produced the sentences in the corpus. Galley et al. (2004) performed a similar process by increasing the maximum number of the derivation. We combine the extracted rules by chart parsing because it is closer to the actual translation process. Using GIZA++, we regarded the intersection of the bi-directional word alignment as an accurate example, which is the first step to extracting the rules. For each grammar type using PS and BS, we vary the maximum length of a sequence ($slimit$) from 1 to 9 to investigate the extraction algorithm. We also restrict the maximum length of a word sequence in the target language ($tlimit$) to 20 by default.

We used Japanese-English parallel corpora provided by the NTCIR-8 PATMT Translation Task (Fujii et al., 2010). The corpora consist of training, development and evaluation corpora. We used the first two as the training corpus for the word alignment, and inspected the development corpus (2,000 sentences) using proposed method. CaboCha[2] is used to obtain the Japanese dependency tree. For each Japanese sentence with the dependency structure, we extracted the rules and tried to generate English sentence by combining the rules. We restricted the maximum number of stored sequences in the chart span to 200 by default.

8.2 Rule extraction

Figure 6 shows the running time of Algorithm 1 for each case using PS and BS. We have graph $f(x) = 0.02x - 0.03$ with $R^2 = 0.99$ for PS, and $f(x) = 0.02x - 0.02$ with $R^2 = 0.99$ for BS where $f(x)$ is the regression function and R^2 is the correlation coefficient. Therefore, the extraction algorithm runs in linear time as we expected.

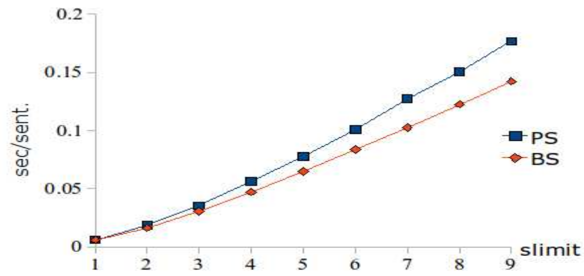


Figure 6: Ellapsed time for the extraction algorithm for each sentence

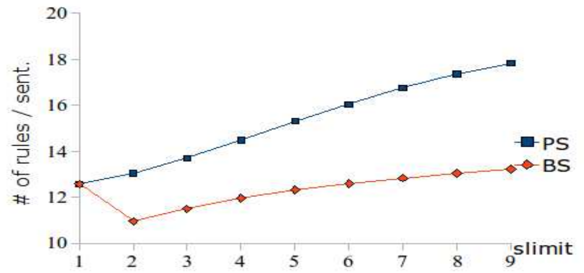


Figure 7: The number of extracted rules for each $slimit$

We also reported the size of the set of rules, which increases linearly with respect to $slimit$ in Figure 7. There is a sudden drop when the number is at $slimit = 2$, and then the number increases. This indicates that the coverage of the re-production also increases.

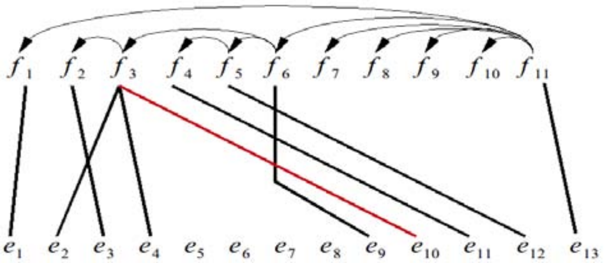
8.3 Rule combination

Figure 8 shows the coverage of the extracted rule using PS and BS. Unfortunately, the coverages are 10% to 50% for PS, and 5% to 35% for BS, when we limit the size to somewhat lower than what we expected. There are several reasons for this:

- Even a single alignment error would cause the failure of the extraction. The example from the corpus below shows that single word alignment error (f_3 and e_{10}) prevents extraction of a production rule, unless we enlarge $slimit$ to

[2] http://www.chasen.org/ taku/software/cabocha/

4. If we remove the incorrect alignment, then $slimit = 1$ is enough to extract a production rule.



- English has much more divergence with Japanese then French. Galley et al. (2004) reported almost 100% coverage between English and French. However, we believe that the language pair we used would have lower coverage than 100% when their method is applied.

- We regard the empirical upper bound of the proposed method as that obtained by unlimited $slimit$. In that case we have coverage of about 60% for both PS and BS. Therefore, we may need other traversal methods such as informed search to broaden the coverage.

9 Conclusion and Future work

The proposed method in this paper addressed a wide range of issues: discontinuous, partial constituent, and non-projective phrases in the source language. We proposed a novel synchronous grammar using the sequences of the traversal order of the dependency tree in the source language. The extraction of phrases takes linear time, and combination takes $O(n^3|G|)$ using a CYK chart parsing algorithm, where $|G|$ is the size of the extracted grammar G. We analyzed our method extensively, which show that the method handles relatively free order language, and both PS and BS are complementary to each other.

The ultimate goal of our proposed method is certainly a syntax-based SMT. In order to develop the decoder, and also improve the proposed method, we will address the remaining issues as follows:

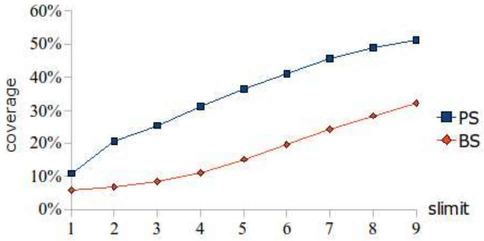


Figure 8: The coverage of the extracted rule

- We regarded the sequences as non-terminals in CFG conceptually. If we use the lexical information directly, however, a data sparseness problem arises as sequences get longer. Therefore we need to generalize the sequence in order for it to be suitable for learning sufficient statistics.

- We utilized only the single best dependency tree, which would not be able to resolve the structural ambiguity. As forest-based rule extraction has been suggested (Mi et al., 2008) in the phrase structure, we will incorporate multiple structures as a compressed one such as a packed-forest.

- There is an another derivation using γ_6, which produces the same target sentence with different rules γ_3 and γ_4:

  $\langle\, X \;,\; X \,\rangle$
  by γ_6: $\Rightarrow \quad \langle\, \langle\, PS_5^5 \;,\; e_1\, PS_1^2\, e_4\, PS_3^4 \,\rangle$
  by γ_4: $\Rightarrow \quad \langle\, \langle\, PS_3^5 \;,\; e_1\, PS_1^2\, e_4 \,[\, e_5 \,]\,\rangle$
  by γ_3: $\Rightarrow \quad \langle\, \langle\, PS_1^5 \;,\; e_1 \,[\, e_2\, e_3 \,]\, e_4 \,[\, e_5 \,]\,\rangle$

  In this case, we reduce one step of the derivation using the production of the non-projective treelet PS_1^2. This indicates that the combination of the minimal rule before decoding, which is commoly used, leads to faster decoding.

Acknowledgments

We appreciate the three anonymous reviewers. This work is supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (MEST) (2009-0075211), in part by the BK 21 project in 2010, and in part by the POSTECH Information Research Laboratories (PIRL) project.

References

Xavier Carreras and Michael Collins. 2009. Non-projective parsing for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 200–209. Association for Computational Linguistics.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 541–548, Morristown, NJ, USA. Association for Computational Linguistics.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 205–208, Morristown, NJ, USA. Association for Computational Linguistics.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 304–3111, Morristown, NJ, USA. Association for Computational Linguistics.

Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro, Terumasa Ehara, Hiroshi Echizen-ya, and Sayori Shimohata. 2010. Overview of the patent translation task at the ntcir-8 workshop. In *Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access*, June.

Michel Galley and Christopher D. Manning. 2010. Accurate non-hierarchical phrase-based translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974, Los Angeles, California, June. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Dekang Lin. 2004. A path-based transfer model for machine translation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 625, Morristown, NJ, USA. 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, June. Association for Computational Linguistics.

Joakim Nivre. 2006. Constraints on non-projective dependency parsing. In *Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 73–80.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: syntactically informed phrasal smt. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 271–279, Morristown, NJ, USA. 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.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2007. A dependency treelet string correspondence model for statistical machine translation. In *StatMT '07: Proceedings of the Second Workshop on Statistical Machine Translation*, pages 40–47, Morristown, NJ, USA. Association for Computational Linguistics.