Translation with Source Constituency and Dependency Trees

Fandong Meng[†§] **Jun Xie**[†] **Linfeng Song**[†§] **Yajuan Lü**[†] **Qun Liu**[‡†]
†Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
§University of Chinese Academy of Sciences
{mengfandong,xiejun,songlinfeng,lvyajuan}@ict.ac.cn
‡Centre for Next Generation Localisation
Faculty of Engineering and Computing, Dublin City University
qliu@computing.dcu.ie

Abstract

We present a novel translation model, which simultaneously exploits the constituency and dependency trees on the source side, to combine the advantages of two types of trees. We take head-dependents relations of dependency trees as backbone and incorporate phrasal nodes of constituency trees as the source side of our translation rules, and the target side as strings. Our rules hold the property of long distance reorderings and the compatibility with phrases. Large-scale experimental results show that our model achieves significantly improvements over the constituency-to-string (+2.45 BLEU on average) and dependency-to-string (+0.91 BLEU on average) models, which only employ single type of trees, and significantly outperforms the state-of-the-art hierarchical phrase-based model (+1.12 BLEU on average), on three Chinese-English NIST test sets.

1 Introduction

In recent years, syntax-based models have become a hot topic in statistical machine translation. According to the linguistic structures, these models can be broadly divided into two categories: constituency-based models (Yamada and Knight, 2001; Graehl and Knight, 2004; Liu et al., 2006; Huang et al., 2006), and dependency-based models (Lin, 2004; Ding and Palmer, 2005; Quirk et al., 2005; Xiong et al., 2007; Shen et al., 2008; Xie et al., 2011). These two kinds of models have their own advantages, as they capture different linguistic phenomena. Constituency trees describe how words and sequences of words combine to form constituents, and constituency-based models show better compatibility with phrases. However, dependency trees describe the grammatical relation between words of the sentence, and represent long distance dependencies in a concise manner. Dependency-based models, such as dependency-to-string model (Xie et al., 2011), exhibit better capability of long distance reorderings.

In this paper, we propose to combine the advantages of source side constituency and dependency trees. Since the dependency tree is structurally simpler and directly represents long distance dependencies, we take dependency trees as the backbone and incorporate constituents to them. Our model employs rules that represent the source side as head-dependents relations which are incorporated with constituency phrasal nodes, and the target side as strings. A head-dependents relation (Xie et al., 2011) is composed of a head and all its dependents in dependency trees, and it encodes phrase pattern and sentence pattern (typically long distance reordering relations). With the advantages of head-dependents relations, the translation rules of our model hold the property of long distance reorderings and the compatibility with phrases.

Our new model (Section 2) extracts rules from word-aligned pairs of source trees (constituency and dependency) and target strings (Section 3), and translate source trees into target strings by employing a bottom-up chart-based algorithm (Section 4). Compared with the constituency-to-string (Liu et al., 2006) and dependency-to-string (Xie et al., 2011) models that only employ a single type of trees, our

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1066–1076,
Seattle, Washington, USA, 18-21 October 2013.

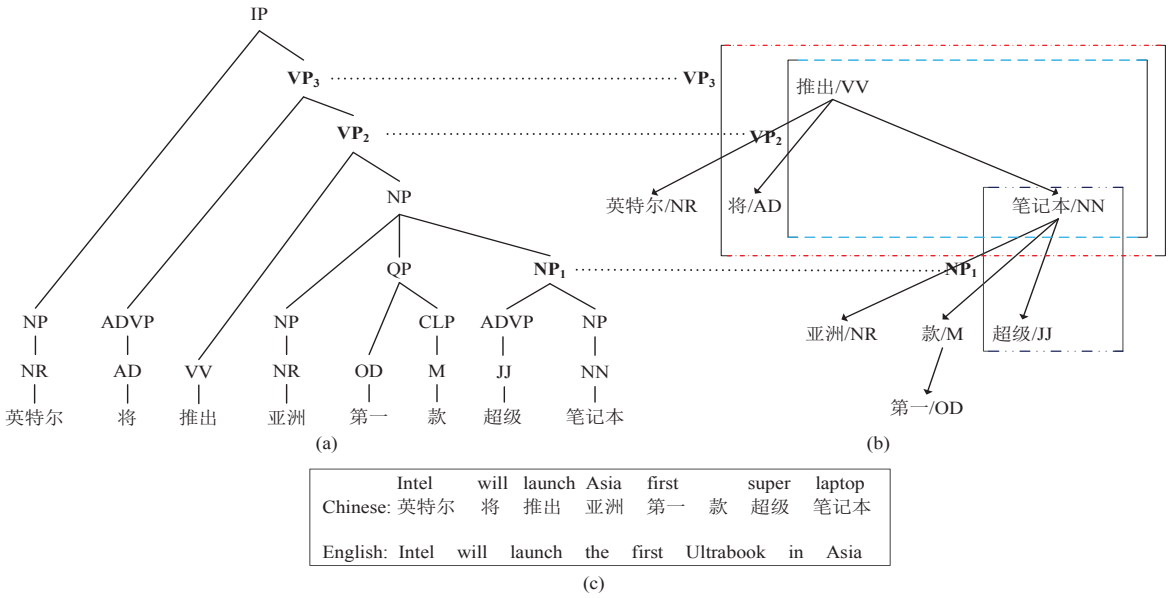

Figure 1: Illustration of phrases that can not be captured by a dependency tree (b) while captured by a constituency tree (a), where the bold phrasal nodes $\mathrm{NP}_1, \mathrm{VP}_2, \mathrm{VP}_3$ indicate the phrases which can not be captured by dependency syntactic phrases. (c) is the corresponding bilingual sentences. The subscripts of phrasal nodes are used for distinguishing the nodes with same phrasal categories.

approach yields encouraging results by exploiting two types of trees. Large-scale experiments (Section 5) on Chinese-English translation show that our model significantly outperforms the state-of-the-art single constituency-to-string model by averaged +2.45 BLEU points, dependency-to-string model by averaged +0.91 BLEU points, and hierarchical phrase-based model (Chiang, 2005) by averaged +1.12 BLEU points, on three Chinese-English NIST test sets.

2 Grammar

We take head-dependents relations of dependency trees as backbone and incorporate phrasal nodes of constituency trees as the source side of our translation rules, and the target side as strings. A head-dependents relation consists of a head and all its dependents in dependency trees, and it can represent long distance dependencies. Incorporating phrasal nodes of constituency trees into head-dependents relations further enhances the compatibility with phrases of our rules. Figure 1 shows an example of phrases which can not be captured by a dependency tree while captured by a constituency tree, such as the bold phrasal nodes NP_1,VP_2 and VP_3. The phrasal node NP_1 in the constituency tree indicates that "超级 笔记本" is a noun phrase and it should be translated as a basic unit, while in the dependency tree it is a non-syntactic phrase. The head-dependents relation in the top level of the dependency tree presents long distance dependencies of the words "英特尔", "将", "推出", and "笔记本" in a concise manner, which is useful for long distance reordering. We adopt this kind of rule representation to hold the property of long distance reorderings and the compatibility with phrases.

Figure 2 shows two examples of our translation rules corresponding to the top level of Figure 1-(b). We can see that r_1 captures a head-dependents relation, while r_2 extends r_1 by incorporating a phrasal node VP_2 to replace the two nodes "推出/VV" and "笔记本/NN". As shown in Figure 1-(b), VP_2 consists of two parts, a head node "推出/VV" and a subtree rooted at the dependent node "笔记本/NN". Therefore, we use VP_2 and the POS tags of the two nodes VV and NN to denote the part covered by VP_2 in r_2, to indicate that the source sequence covered by VP_2 can be translated by a bilingual phrase. Since VP_2 covers a head node "推出/VV", we represent r_2 by constructing a new head node

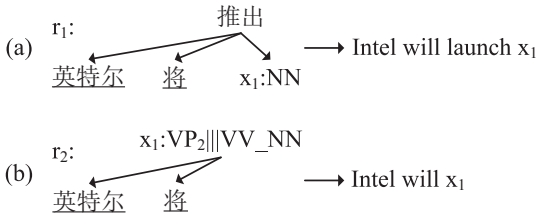

Figure 2: Two examples of our translation rules corresponding to the top level of Figure 1-(b). r_1 captures a head-dependents relation, and r_2 extends r_1 by incorporating a phrasal node VP_2. "x_1:NN" indicates a substitution site which can be replaced by a subtree whose root has POS tag "NN". "x_1:VP_2|||VV_NN" indicates a substitution site which can be replaced by a source phrase covered by a phrasal node VP (the phrasal node consists of two dependency nodes with POS tag VV and NN, respectively). The underline denotes a leaf node.

VP_2|||VV_NN. For simplicity, we use a shorten form CHDR to represent the head-dependents relations with/without constituency phrasal nodes.

Formally, our grammar G is defined as a 5-tuple $G = \langle \Sigma, N_c, N_d, \Delta, R \rangle$, where Σ is a set of source language terminals, N_c is a set of constituency phrasal categories, N_d is a set of categories (POS tags) for the terminals in Σ, Δ is a set of target language terminals, and R is a set of translation rules that include bilingual phrases for translating source language terminals and CHDR rules for translation and reordering. A CHDR rule is represented as a triple $\langle t, s, \sim \rangle$, where:

- t is CHDR with each node labeled by a terminal from Σ or a variable from a set $X = \{x_1, x_2, \cdots\}$ constrained by a terminal from Σ or a category from N_d or a joint category (constructed by the categories from N_c and N_d);
- $s \in (X \cup \Delta)$ denotes the target side string;
- $\sim$ denotes one-to-one links between nonterminals in t and variables in s.

We use the lexicon dependency grammar (Hellwig, 2006) which adopts a bracket representation to express the head-dependents relation and CHDR. For example, the left-hand sides of r_1 and r_2 in Figure 2 can be respectively represented as follows:

(英特尔) (将) 推出 (x_1:NN)

(英特尔) (将) x_1:VP_2|||VV_NN

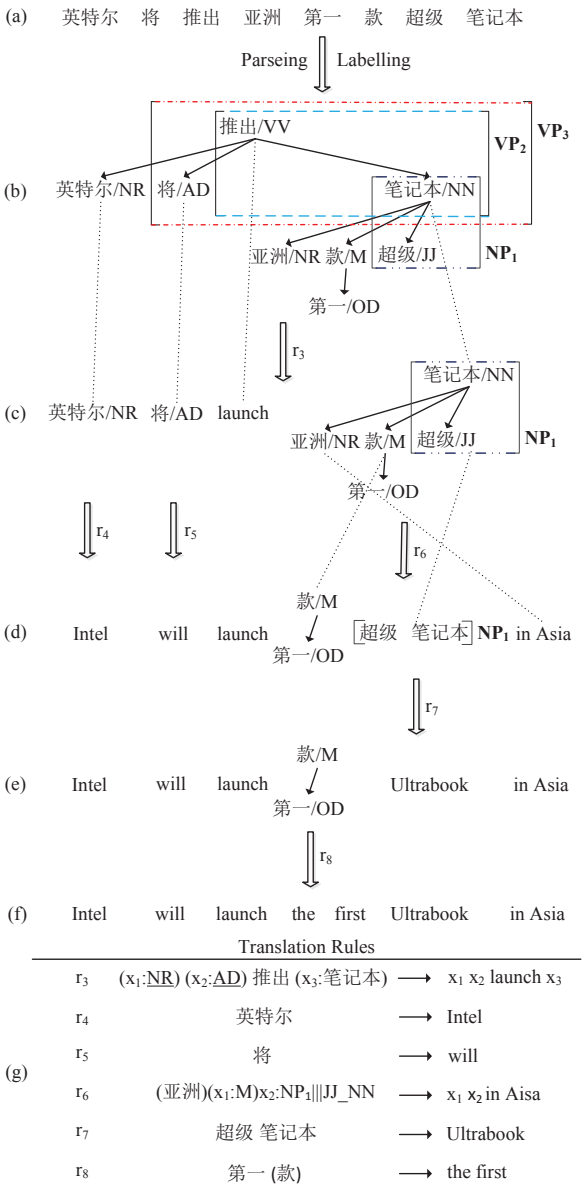

Figure 3: An example derivation of translation. (g) lists all the translation rules. r_3, r_6 and r_8 are CHDR rules, while r_4, r_5 and r_7 are bilingual phrases, which are used for translating source terminals. The dash lines indicate the reordering when employing a translation rule.

The formalized presentation of r_2 in Figure 2-(b):

t = (英特尔) (将) x_1:VP_2|||VV_NN

s = Intel will x_1

$\sim$= x_1:VP_2|||VV_NN $\leftrightarrow$ x_1

where the underline indicates a leaf node.

Figure 3 gives an example of the translation derivation in our model, with the translation rules

listed in (g). r_3, r_6 and r_8 are CHDR rules, while r_4, r_5 and r_7 are bilingual phrases, which are used for translating source language terminals. Given a sentence to translate in (a), we first parse it into a constituency tree and a dependency tree, then label the phrasal nodes from the constituency tree to the dependency tree, and yield (b). Then, we translate it into a target string by the following steps. At the root node, we apply rule r_3 to translate the top level head-dependents relation and results in four unfinished substructures and target strings in (c). From (c) to (d), there are three steps (one rule for one step). We use r_4 to translate "英特尔" to "Intel", r_5 to translate "将" to "will", and r_6 to translate the rightmost unfinished part. Then, we apply r_7 to translate the phrase "超级 笔记本" to "Ultrabook", and yield (e). Finally, we apply r_8 to translate the last fragment to "the first", and get the final result (f).

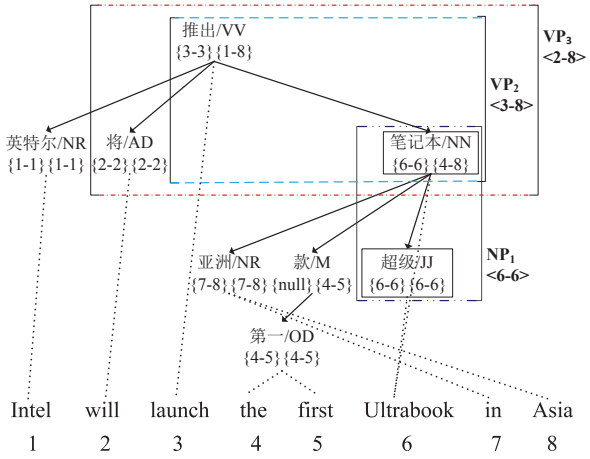

Figure 4: An annotated dependency tree. Each node is annotated with two spans, the former is node span and the latter subtree span. The fragments covered by phrasal nodes are annotated with phrasal spans. The nodes denoted by the solid line box are not nsp consistent.

3 Rule Extraction

In this section, we describe how to extract rules from a set of 4-tuples $\langle C, T, S, A\rangle$, where C is a source constituency tree, T is a source dependency tree, S is a target side sentence, and A is an word alignment relation between T/C and S. We extract CHDR rules from each 4-tuple $\langle C, T, S, A\rangle$ based on GHKM algorithm (Galley et al., 2004) with three steps:

1. Label the dependency tree with phrasal nodes from the constituency tree, and annotate alignment information to the phrasal nodes labeled dependency tree (Section 3.1).

2. Identify acceptable CHDR fragments from the annotated dependency tree for rule induction (Section 3.2).

3. Induce a set of lexicalized and generalized CHDR rules from the acceptable fragments (Section 3.3).

3.1 Annotation

Given a 4-tuple $\langle C, T, S, A\rangle$, we first label phrasal nodes from the constituency tree C to the dependency tree T, which can be easily accomplished by phrases mapping according to the common covered source sequences. As dependency trees can capture some phrasal information by dependency syntactic phrases, in order to complement the information that dependency trees can not capture, we only label the phrasal nodes that cover dependency non-syntactic phrases.

Then, we annotate alignment information to the phrasal nodes labeled dependency tree T, as shown in Figure 4. For description convenience, we make use of the notion of spans (Fox, 2002; Lin, 2004). Given a node n in the source phrasal nodes labeled T with word alignment information, the spans of n induced by the word alignment are consecutive sequences of words in the target sentence. As shown in Figure 4, we annotate each node n of phrasal nodes labeled T with two attributes: *node span* and *subtree span*; besides, we annotate *phrasal span* to the parts covered by phrasal nodes in each subtree rooted at n. The three types of spans are defined as follows:

Definition 1 *Given a node n, its **node span** $nsp(n)$ is the consecutive target word sequence aligned with the node n.*

Take the node "亚洲/NR" in Figure 4 for example, nsp(亚洲/NR)={7-8}, which corresponds to the target words "in" and "Asia".

Definition 2 *Given a subtree T' rooted at n, the **subtree span** $tsp(n)$ of n is the consecutive target word sequence from the lower bound of the nsp of*

all nodes in T' to the upper bound of the same set of spans.

For instance, tsp(笔记本/NN)={4-8}, which corresponds to the target words "the first Ultrabook in Asia", whose indexes are from 4 to 8.

Definition 3 *Given a fragment f covered by a phrasal node, the* ***phrasal span*** $psp(f)$ *of f is the consecutive target word sequence aligned with source string covered by f.*

For example, $psp(\text{VP}_2)$=⟨3-8⟩, which corresponds to the target word sequence "launch the first Ultrabook in Asia".

We say nsp, tsp and psp are *consistent* according to the notion in the phrase-based model (Koehn et al., 2003). For example, nsp(亚洲/NR), tsp(笔记本/NN) and $psp(\text{NP}_1)$ are consistent while nsp(超级/JJ) and nsp(笔记本/NN) are not consistent.

The annotation can be achieved by a single postorder transversal of the phrasal nodes labeled dependency tree. For simplicity, we call the annotated phrasal nodes labeled dependency tree *annotated dependency tree*. The extraction of bilingual phrases (including the translation of head node, dependency syntactic phrases and the fragment covered by a phrasal node) can be readily achieved by the algorithm described in Koehn et al., (2003). In the following, we focus on CHDR rules extraction.

3.2 Acceptable Fragments Identification

Before present the method of acceptable fragments identification, we give a brief description of CHDR fragments. A CHDR fragment is an annotated fragment that consists of a source head-dependents relation with/without constituency phrasal nodes, a target string and the word alignment information between the source and target side. We identify the acceptable CHDR fragments that are suitable for rule induction from the annotated dependency tree. We divide the acceptable CHDR fragments into two categories depending on whether the fragments contain phrasal nodes. If an acceptable CHDR fragment does not contain phrasal nodes, we call it *CHDR-normal fragment*, otherwise *CHDR-phrasal fragment*. Given a CHDR fragment F rooted at n, we say F is acceptable if it satisfies any one of the following properties:

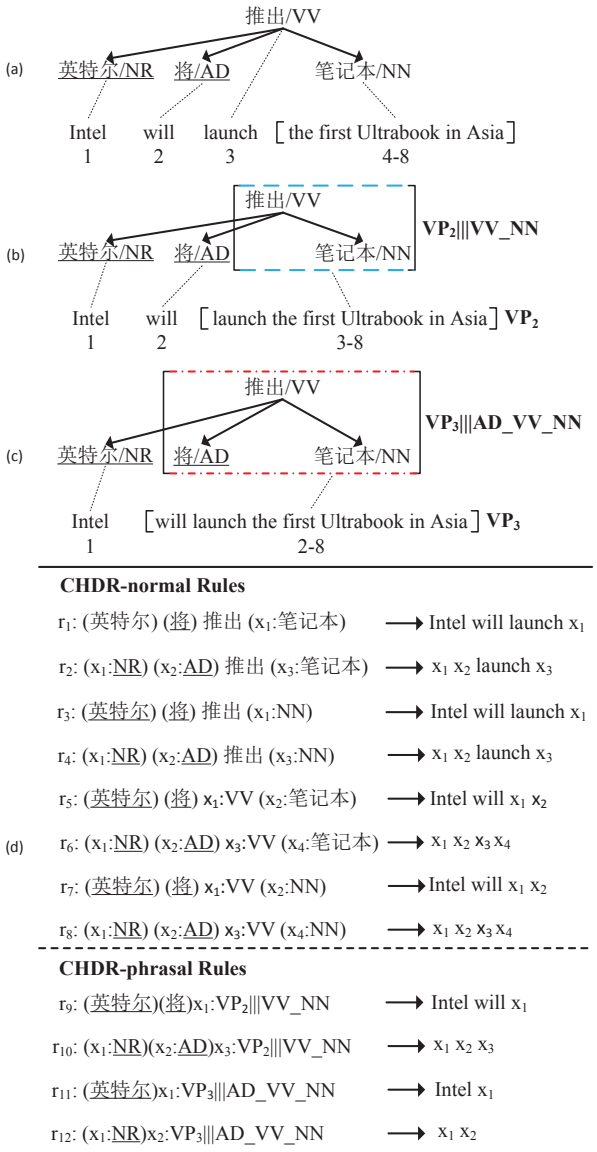

Figure 5: Examples of a CHDR-normal fragment (a), two CHDR-phrasal fragments (b) and (c) that are identified from the top level of the annotated dependency tree in Figure 4, and the corresponding CHDR rules (d) induced from (a), (b) and (c). The underline denotes a leaf node.

1. Without phrasal nodes, the node span of the root n is consistent and the subtree spans of n's all dependents are consistent. For example, Figure 5-(a) shows a CHDR-normal fragment that identified from the top level of the annotated dependency tree in Figure 4, since the nsp(推出/VV), tsp(英特尔/NR), tsp(将/AD) and tsp(笔记本/NN) are consistent.

2. With phrasal nodes, the phrasal spans of phrasal nodes are consistent; and for the other nodes, the node span of head (if it is not covered by any phrasal node) is consistent, and the subtree spans of dependents are consistent. For instance, Figure 5-(b) and (c) show two CHDR-phrasal fragments identified from the top level of Figure 4. In Figure 5-(b), $psp(\text{VP}_2)$, tsp(英特尔/NR) and tsp(将/AD) are consistent. In Figure 5-(c), $psp(\text{VP}_3)$ and tsp(英特尔/NR) are consistent.

The identification of acceptable fragments can be achieved by a single postorder transversal of the annotated dependency tree. Typically, each acceptable fragment contains at most three types of nodes: head node, head of the related CHDR; internal nodes, internal nodes of the related CHDR except head node; leaf nodes, leaf nodes of the related CHDR.

3.3 Rule Induction

From each acceptable CHDR fragment, we induce a set of lexicalized and generalized CHDR rules. We induce CHDR-normal rules and CHDR-phrasal rules from CHDR-normal fragments and CHDR-phrasal fragments, respectively.

We first induce a lexicalized form of CHDR rule from an acceptable CHDR fragment:

1. For a CHDR-normal fragment, we first mark the internal nodes as substitution sites. This forms the input of a CHDR-normal rule. Then we generate the target string according to the node span of the head and the subtree spans of the dependents, and turn the word sequences covered by the internal nodes into variables. This forms the output of a lexicalized CHDR-normal rule.

2. For a CHDR-phrasal fragment, we first mark the internal nodes and the phrasal nodes as substitution sites. This forms the input of a CHDR-phrasal rule. Then we construct the output of the CHDR-phrasal rule in almost the same way with constructing CHDR-normal rules, except that we replace the target sequences covered by the internal nodes and the phrasal nodes with variables.

For example, rule r_1 in Figure 5-(d) is a lexicalized CHDR-normal rule induced from the CHDR-normal fragment in Figure 5-(a). r_9 and r_{11} are CHDR-phrasal rules induced from the CHDR-phrasal fragment in Figure 5-(b) and Figure 5-(c) respectively. As we can see, these CHDR-phrasal rules are partially unlexicalized.

To alleviate the sparseness problem, we generalize the lexicalized CHDR-normal rules and partially unlexicalized CHDR-phrasal rules with unlexicalized nodes by the method proposed in Xie et al., (2011). As the modification relations between head and dependents are determined by the edges, we can replace the lexical word of each node with its category (POS tag) and obtain new head-dependents relations with unlexicalized nodes keeping the same modification relations. We generalize the rule by simultaneously turn the nodes of the same type (head, internal, leaf) into their categories. For example, CHDR-normal rules $r_2 \sim r_7$ are generalized from r_1 in Figure 5-(d). Besides, r_{10} and r_{12} are the corresponding generalized CHDR-phrasal rules. Actually, our CHDR rules are the superset of head-dependents relation rules in Xie et al., (2011). CHDR-normal rules are equivalent with the head-dependents relation rules and the CHDR-phrasal rules are the extension of these rules. For convenience of description, we use the subscript to distinguish the phrasal nodes with the same category, such as VP_2 and VP_3. In actual operation, we use VP instead of VP_2 and VP_3.

We handle the unaligned words of the target side by extending the node spans of the lexicalized head and leaf nodes, and the subtree spans of the lexicalized dependents, on both left and right directions. This procedure is similar with the method of Och and Ney, (2004). During this process, we might obtain $m(m \geq 1)$ CHDR rules from an acceptable fragment. Each of these rules is assigned with a fractional count $1/m$. We take the extracted rule set as observed data and make use of relative frequency estimator to obtain the translation probabilities $P(t|s)$ and $P(s|t)$.

4 Decoding and the Model

Following Och and Ney, (2002), we adopt a general loglinear model. Let d be a derivation that convert a

source phrasal nodes labeled dependency tree into a target string e. The probability of d is defined as:

$$P(d) \propto \prod_i \phi_i(d)^{\lambda_i} \quad (1)$$

where ϕ_i are features defined on derivations and λ_i are feature weights. In our experiments of this paper, the features are used as follows:

- CHDR rules translation probabilities $P(t|s)$ and $P(s|t)$, and CHDR rules lexical translation probabilities $P_{lex}(t|s)$ and $P_{lex}(s|t)$;
- bilingual phrases translation probabilities $P_{bp}(t|s)$ and $P_{bp}(s|t)$, and bilingual phrases lexical translation probabilities $P_{bplex}(t|s)$ and $P_{bplex}(s|t)$;
- rule penalty $exp(-1)$;
- pseudo translation rule penalty $exp(-1)$;
- target word penalty $exp(|e|)$;
- language model $P_{lm}(e)$.

We have twelve features in our model. The values of the first four features are accumulated on the CHDR rules and the next four features are accumulated on the bilingual phrases. We also use a pseudo translation rule (constructed according to the word order of head-dependents relation) as a feature to guarantee the complete translation when no matched rules can be found during decoding.

Our decoder is based on bottom-up chart-based algorithm. It finds the best derivation that convert the input phrasal nodes labeled dependency tree into a target string among all possible derivations. Given the source constituency tree and dependency tree, we first generate phrasal nodes labeled dependency tree T as described in Section 3.1, then the decoder transverses each node in T by postorder. For each node n, it enumerates all instances of CHDR rooted at n, and checks the rule set for matched translation rules. A larger translation is generated by substituting the variables in the target side of a translation rule with the translations of the corresponding dependents. Cube pruning (Chiang, 2007; Huang and Chiang, 2007) is used to find the k-best items with integrated language model for each node.

To balance the performance and speed of the decoder, we limit the search space by reducing the number of translation rules used for each node. There are two ways to limit the rule table size: by a fixed limit (rule-limit) of how many rules are retrieved for each input node, and by a threshold (rule-threshold) to specify that the rule with a score lower than β times of the best score should be discarded. On the other hand, instead of keeping the full list of candidates for a given node, we keep a top-scoring subset of the candidates. This can also be done by a fixed limit (stack-limit) and a threshold (stack-threshold).

5 Experiments

We evaluated the performance of our model by comparing with hierarchical phrase-based model (Chiang, 2007), constituency-to-string model (Liu et al., 2006) and dependency-to-string model (Xie et al., 2011) on Chinese-English translation. First, we describe data preparation (Section 5.1) and systems (Section 5.2). Then, we validate that our model significantly outperforms all the other baseline models (Section 5.3). Finally, we give detail analysis (Section 5.4).

5.1 Data Preparation

Our training data consists of 1.25M sentence pairs extracted from LDC [1] data. We choose NIST MT Evaluation test set 2002 as our development set, NIST MT Evaluation test sets 2003 (MT03), 2004 (MT04) and 2005 (MT05) as our test sets. The quality of translations is evaluated by the case insensitive NIST BLEU-4 metric [2].

We parse the source sentences to constituency trees (without binarization) and projective dependency trees with Stanford Parser (Klein and Manning, 2002). The word alignments are obtained by running GIZA++ (Och and Ney, 2003) on the corpus in both directions and using the "grow-diag-final-and" balance strategy (Koehn et al., 2003). We get bilingual phrases from word-aligned data with algorithm described in Koehn et al. (2003) by running Moses Toolkit [3]. We apply SRI Language Modeling Toolkit (Stolcke and others, 2002) to train a 4-gram

[1] Including LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

[2] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl

[3] http://www.statmt.org/moses/

System	Rule #	MT03	MT04	MT05	Average
Moses-chart	116.4M	34.65	36.47	34.39	35.17
cons2str	25.4M+32.5M	33.14	35.12	33.27	33.84
dep2str	19.6M+32.5M	34.85	36.57	34.72	35.38
consdep2str	23.3M+32.5M	**35.57***	**37.68***	**35.62***	**36.29**

Table 1: Statistics of the extracted rules on training data and the BLEU scores (%) on the test sets of different systems. The "+" denotes that the rules are composed of syntactic translation rules and bilingual phrases (32.5M). The "*" denotes that the results are significantly better than all the other systems ($p<0.01$).

language model with modified Kneser-Ney smoothing on the Xinhua portion of the English Gigaword corpus. We make use of the standard MERT (Och, 2003) to tune the feature weights in order to maximize the system's BLEU score on the development set. The statistical significance test is performed by *sign-test* (Collins et al., 2005).

5.2 Systems

We take the open source hierarchical phrase-based system *Moses-chart* (with default configuration), our in-house constituency-to-string system *cons2str* and dependency-to-string system *dep2str* as our baseline systems.

For *cons2str*, we follow Liu et al., (Liu et al., 2006) to strict that the height of a rule tree is no greater than 3 and phrase length is no greater than 7. To keep consistent with our proposed model, we implement the dependency-to-string model (Xie et al., 2011) with GHKM (Galley et al., 2004) rule extraction algorithm and utilize bilingual phrases to translate source head node and dependency syntactic phrases. Our *dep2str* shows comparable performance with Xie et al., (2011), which can be seen by comparing with the results of hierarchical phrase-based model in our experiments. For *dep2str* and our proposed model *consdep2str*, we set rule-threshold and stack-threshold to 10^{-3}, rule-limit to 100, stack-limit to 300, and phrase length limit to 7.

5.3 Experimental Results

Table 1 illustrates the translation results of our experiments. As we can see, our *consdep2str* system has gained the best results on all test sets, with +1.12 BLEU points higher than *Moses-chart*, +2.45 BLEU points higher than *cons2str*, and +0.91 BLEU points higher than *dep2str*, averagely on MT03, MT04 and MT05. Our model significantly outperforms all the other baseline models, with $p<0.01$ on statistical significance test *sign-test* (Collins et al., 2005). By exploiting two types of trees on source side, our model gains significant improvements over constituency-to-string and dependency-to-string models, which employ single type of trees.

Table 1 also lists the statistical results of rules extracted from training data by different systems. According to our statistics, the number of rules extracted by our *consdep2str* system is about 18.88% larger than *dep2str*, without regard to the 32.5M bilingual phrases. The extra rules are CHDR-phrasal rules, which can bring in BLEU improvements by enhancing the compatibility with phrases. We will conduct a deep analysis in the next sub-section.

5.4 Analysis

In this section, we first illustrate the influence of CHDR-phrasal rules in our *consdep2str* model. We calculate the proportion of 1-best translations in test sets that employ CHDR-phrasal rules, and we call this proportion "*CHDR-phrasal Sent.*". Besides, the proportion of CHDR-phrasal rules in all CHDR rules is calculated in these translations, and we call this proportion "*CHDR-phrasal Rule*". Table 2 lists the using of CHDR-phrasal rules on test sets, showing that *CHDR-phrasal Sent.* on all test sets are higher than 50%, and *CHDR-phrasal Rule* on all three test sets are higher than 10%. These results indicate that CHDR-phrasal rules do play a role in decoding.

Furthermore, we compare some actual translations of our test sets generated by *cons2str*, *dep2str* and *consdep2str* systems, as shown in Figure 6. In the first example, the Chinese input holds long distance dependencies "联合国 已经 对 ... 加诸于 ... 表示 关切", which correspond to the sentence pattern "noun+adverb+prepositional

System	MT03	MT04	MT05
CHDR-phrasal Sent.	50.71	61.80	56.19
CHDR-phrasal Rule	10.53	13.55	10.83

Table 2: The proportion (%) of 1-best translations that employs CHDR-phrasal rules (CHDR-phrasal Sent.) and the proportion (%) of CHDR-phrasal rules in all CHDR rules in these translations (CHDR-phrasal Rule).

phrase+verb+noun". *Cons2str* gives a bad result with wrong global reordering, while our *consdep2str* system gains an almost correct result since we capture this pattern by CHDR-normal rules. In the second example, we can see that the Chinese phrase "再次 出现" is a non-syntactic phrase in the dependency tree, and this phrase can not be captured by head-dependents relation rules in Xie et al., (2011), thus can not be translated as one unit. Since we encode constituency phrasal nodes to the dependency tree, "再次 出现" is labeled by a phrasal node "VP" (means verb phrase), which can be captured by our CHDR-phrasal rules and translated into the correct result "reemergence" with bilingual phrases.

By combining the merits of constituency and dependency trees, our *consdep2str* model learns CHDR-normal rules to acquire the property of long distance reorderings and CHDR-phrasal rules to obtain good compatibility with phrases.

6 Related Work

In recent years, syntax-based models have witnessed promising improvements. Some researchers make efforts on constituency-based models (Graehl and Knight, 2004; Liu et al., 2006; Huang et al., 2006; Zhang et al., 2007; Mi et al., 2008; Liu et al., 2009; Liu et al., 2011; Zhai et al., 2012). Some works pay attention to dependency-based models (Lin, 2004; Ding and Palmer, 2005; Quirk et al., 2005; Xiong et al., 2007; Shen et al., 2008; Xie et al., 2011). These models are based on single type of trees.

There are also some approaches combining merits of different structures. Marton and Resnik (2008) took the source constituency tree into account and added soft constraints to the hierarchical phrase-based model (Chiang, 2005). Cherry (2008) utilized dependency tree to add syntactic cohesion to the phrased-based model. Mi and Liu, (2010) proposed a constituency-to-dependency translation model, which utilizes constituency forests on the source side to direct the translation, and dependency trees on the target side to ensure grammaticality. Feng et al. (2012) presented a hierarchical chunk-to-string translation model, which is a compromise between the hierarchical phrase-based model and the constituency-to-string model. Most works make effort to introduce linguistic knowledge into the phrase-based model and hierarchical phrase-based model with constituency trees. Only the work proposed by Mi and Liu, (2010) utilized constituency and dependency trees, while their work applied two types of trees on two sides.

Instead, our model simultaneously utilizes constituency and dependency trees on the source side to direct the translation, which is concerned with combining the advantages of two types of trees in translation rules to advance the state-of-the-art machine translation.

7 Conclusion

In this paper, we present a novel model that simultaneously utilizes constituency and dependency trees on the source side to direct the translation. To combine the merits of constituency and dependency trees, our model employs head-dependents relations incorporating with constituency phrasal nodes. Experimental results show that our model exhibits good performance and significantly outperforms the state-of-the-art constituency-to-string, dependency-to-string and hierarchical phrase-based models. For the first time, source side constituency and dependency trees are simultaneously utilized to direct the translation, and the model surpasses the state-of-the-art translation models.

Since constituency tree binarization can lead to more constituency-to-string rules and syntactic phrases in rule extraction and decoding, which improve the performance of constituency-to-string systems, for future work, we would like to do research on encoding binarized constituency trees to dependency trees to improve translation performance.

Acknowledgments

The authors were supported by National Natural Science Foundation of China (Contracts 61202216),

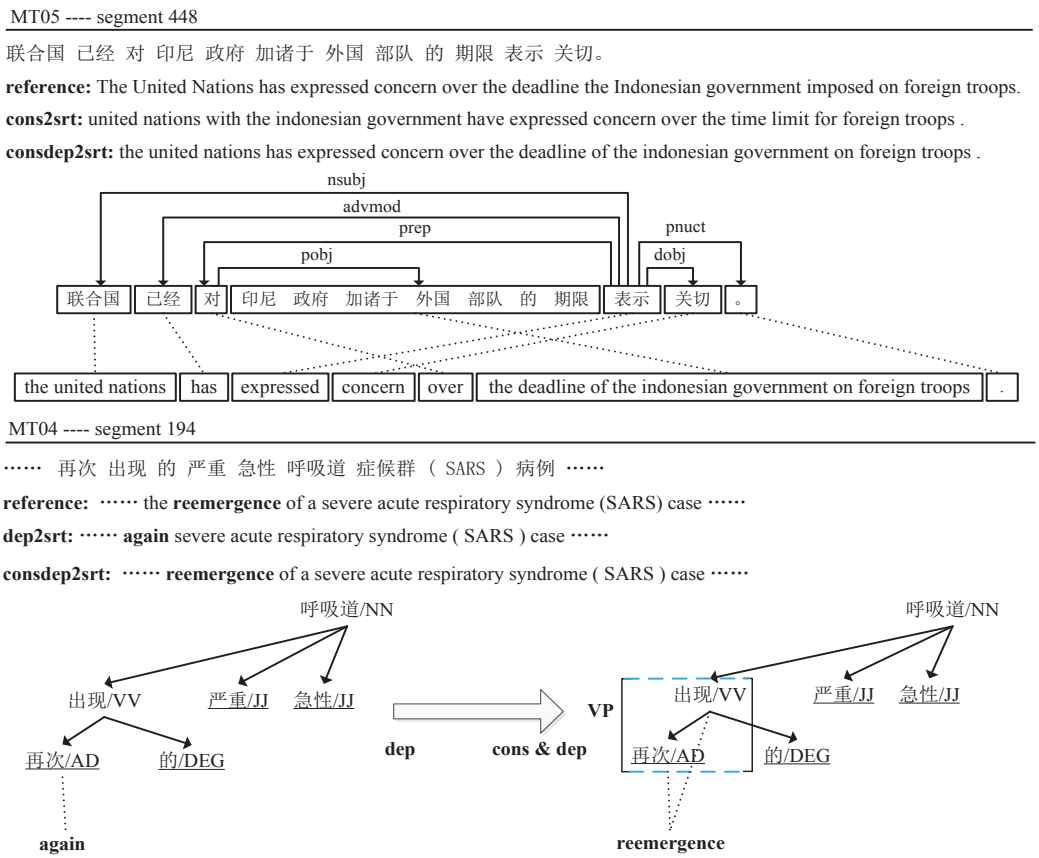

Figure 6: Actual examples translated by the *cons2str*, *dep2str* and *consdep2str* systems.

863 State Key Project (No. 2011AA01A207), and National Key Technology R&D Program (No. 2012BAH39B03), Key Project of Knowledge Innovation Program of Chinese Academy of Sciences (No. KGZD-EW-501). Qun Liu's work was partially supported by Science Foundation Ireland (Grant No. 07/CE/I1142) as part of the CNGL at Dublin City University. Sincere thanks to the anonymous reviewers for their thorough reviewing and valuable suggestions. We appreciate Haitao Mi, Zhaopeng Tu and Anbang Zhao for insightful advices in writing.

References

Colin Cherry. 2008. Cohesive phrase-based decoding for statistical machine translation. In *ACL*, pages 72–80.

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.

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

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 531–540.

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

Yang Feng, Dongdong Zhang, Mu Li, Ming Zhou, and Qun Liu. 2012. Hierarchical chunk-to-string translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 950–958.

Heidi J Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 304–3111.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule. In *Pro-*

ceedings of HLT/NAACL, volume 4, pages 273–280. Boston.

Jonathan Graehl and Kevin Knight. 2004. Training tree transducers. In *Proc. HLT-NAACL*, pages 105–112.

Peter Hellwig. 2006. Parsing with dependency grammars. *An International Handbook of Contemporary Research*, 2:1081–1109.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Annual Meeting-Association For Computational Linguistics*, volume 45, pages 144–151.

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

Dan Klein and Christopher D Manning. 2002. Fast exact inference with a factored model for natural language parsing. In *Advances in neural information processing systems*, volume 15, pages 3–10.

Philipp Koehn, Franz Josef 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.

Dekang Lin. 2004. A path-based transfer model for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 625–630.

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 the 44th annual meeting of the Association for Computational Linguistics*, pages 609–616.

Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. 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: Volume 2-Volume 2*, pages 558–566.

Yang Liu, Qun Liu, and Yajuan Lü. 2011. Adjoining tree-to-string translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1278–1287.

Yuval Marton and Philip Resnik. 2008. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of ACL-08: HLT*, pages 1003–1011.

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.

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

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

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

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational linguistics*, 30(4):417–449.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167.

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

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.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing*, volume 2, pages 901–904.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 216–226.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2007. A dependency treelet string correspondence model for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 40–47.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530.

Feifei Zhai, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2012. Tree-based translation without using parse trees. In *Proceedings of COLING 2012*, pages 3037–3054.

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