A Novel Dependency-to-String Model for Statistical Machine Translation

Jun Xie, Haitao Mi and Qun Liu
Key Laboratory of Intelligent Information Processiong
Institute of Computing Technology
Chinese Academy of Sciences
P.O. Box 2704, Beijing 100190, China
{junxie,htmi,liuqun}@ict.ac.cn

Abstract

Dependency structure, as a first step towards semantics, is believed to be helpful to improve translation quality. However, previous works on dependency structure based models typically resort to insertion operations to complete translations, which make it difficult to specify ordering information in translation rules. In our model of this paper, we handle this problem by directly specifying the ordering information in head-dependents rules which represent the source side as head-dependents relations and the target side as strings. The head-dependents rules require only substitution operation, thus our model requires no heuristics or separate ordering models of the previous works to control the word order of translations. Large-scale experiments show that our model performs well on long distance reordering, and outperforms the state-of-the-art constituency-to-string model (+1.47 BLEU on average) and hierarchical phrase-based model (+0.46 BLEU on average) on two Chinese-English NIST test sets without resort to phrases or parse forest. For the first time, a source dependency structure based model catches up with and surpasses the state-of-the-art translation models.

1 Introduction

Dependency structure represents the grammatical relations that hold between the words in a sentence. It encodes semantic relations directly, and has the best inter-lingual phrasal cohesion properties (Fox, 2002). Those attractive characteristics make it possible to improve translation quality by using dependency structures.

Some researchers pay more attention to use dependency structure on the target side. (Shen et al., 2008) presents a string-to-dependency model, which restricts the target side of each hierarchical rule to be a well-formed dependency tree fragment, and employs a dependency language model to make the output more grammatically. This model significantly outperforms the state-of-the-art hierarchical phrase-based model (Chiang, 2005). However, those string-to-tree systems run slowly in cubic time (Huang et al., 2006).

Using dependency structure on the source side is also a promising way, as tree-based systems run much faster (linear time vs. cubic time, see (Huang et al., 2006)). Conventional dependency structure based models (Lin, 2004; Quirk et al., 2005; Ding and Palmer, 2005; Xiong et al., 2007) typically employ both substitution and insertion operation to complete translations, which make it difficult to specify ordering information directly in the translation rules. As a result, they have to resort to either heuristics (Lin, 2004; Xiong et al., 2007) or separate ordering models (Quirk et al., 2005; Ding and Palmer, 2005) to control the word order of translations.

In this paper, we handle this problem by directly specifying the ordering information in head-dependents rules that represent the source side as head-dependents relations and the target side as string. The head-dependents rules have only one substitution operation, thus we don't face the problems appeared in previous work and get rid of the

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 216–226,
Edinburgh, Scotland, UK, July 27–31, 2011.

heuristics and ordering model. To alleviate data sparseness problem, we generalize the lexicalized words in head-dependents relations with their corresponding categories.

In the following parts, we first describe the motivation of using head-dependents relations (Section 2). Then we formalize our grammar (Section 3), present our rule acquisition algorithm (Section 4), our model (Section 5) and decoding algorithm (Section 6). Finally, large-scale experiments (Section 7) show that our model exhibits good performance on long distance reordering, and outperforms the state-of-the-art tree-to-string model (+1.47 BLEU on average) and hierarchical phrase-based model (+0.46 BLEU on average) on two Chinese-English NIST test sets. For the first time, a source dependency tree based model catches up with and surpasses the state-of-the-art translation models.

2 Dependency Structure and Head-Dependents Relation

2.1 Dependency Sturcture

A dependency structure for a sentence is a directed acyclic graph with words as nodes and modification relations as edges. Each edge direct from a head to a dependent. Figure 1 (a) shows an example dependency structure of a Chinese sentence.

2010年 FIFA 世界杯 在 南非 成功 举行

2010 FIFA [World Cup] in/at [South Africa] successfully hold

Each node is annotated with the part-of-speech (POS) of the related word.

For convenience, we use the lexicon dependency grammar (Hellwig, 2006) which adopts a bracket representation to express a projective dependency structure. The dependency structure of Figure 1 (a) can be expressed as:

((2010年) (FIFA) 世界杯) (在(南非)) (成功) 举行

where the lexicon in brackets represents the dependents, while the lexicon out the brackets is the head.

To construct the dependency structure of a sentence, the most important thing is to establish dependency relations and distinguish the head from the dependent. Here are some criteria (Zwicky, 1985; Hudson, 1990) for identifying a syntactic relation between a *head* and a *dependent* between a head-dependent pair:

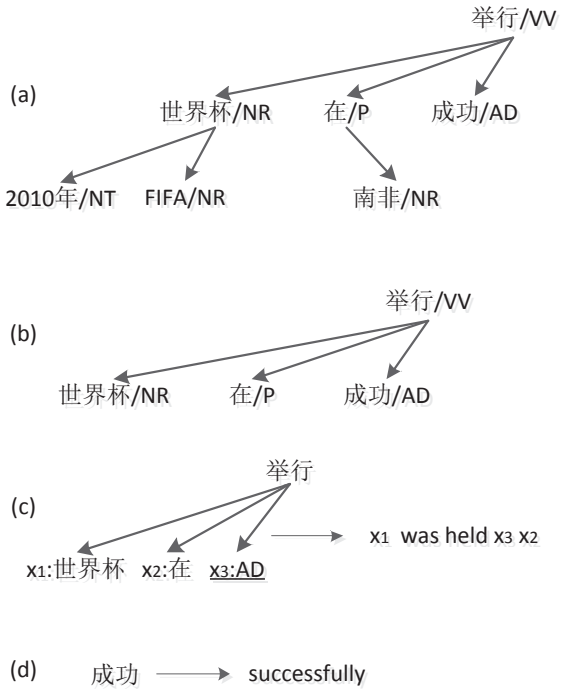

Figure 1: Examples of dependency structure (a), head-dependents relation (b), head-dependents rule (r_1 of Figure 2) and head rule (d). Where "x_1:世界杯" and "x_2:在" indicate substitution sites which can be replaced by a subtree rooted at "世界杯" and "在" respectively. "x_3:AD"indicates a substitution site that can be replaced by a subtree whose root has part-of-speech "AD". The underline denotes a leaf node.

1. *head* determines the syntactic category of *C*, and can often replace *C*;
2. *head* determines the semantic category of *C*; *dependent* gives semantic specification.

2.2 Head-Dependents Relation

A head-dependents relation is composed of a head and all its dependents as shown in Figure 1(b).

Since all the head-dependent pairs satisfy criteria 1 and 2, we can deduce that a head-dependents relation *L* holds the property that *the head determines the syntactic and semantic categories of L, and can often replace L.* Therefore, we can recur-

sively replace the bottom level head-dependent relations of a dependency structure with their heads until the root. This implies an representation of the generation of a dependency structure on the basis of head-dependents relation.

Inspired by this, we represent the translation rules of our dependency-to-string model on the foundation of head-dependents relations.

3 Dependency-to-String Grammar

Figure 1 (c) and (d) show two examples of the translation rules used in our dependency-to-string model. The former is an example of *head-dependent rules* that represent the source side as head-dependents relations and act as both translation rules and reordering rules. The latter is an example of *head rules* which are used for translating words.

Formally, a dependency-to-string grammar is defined as a tuple $\langle \Sigma, N, \Delta, R \rangle$, where Σ is a set of source language terminals, N is a set of categories for the terminals in Σ , Δ is a set of target language terminals, and R is a set of translation rules. A rule r in R is a tuple $\langle t, s, \phi \rangle$, where:

- t is a node labeled by terminal from Σ; or a head-dependents relation of the source dependency structures, with each node labeled by a terminal from Σ or a variable from a set $X = \{x_1, x_2, ...\}$ constrained by a terminal from Σ or a category from N;
- $s \in (X \cup \Delta)^*$ is the target side string;
- ϕ is a one-to-one mapping from nonterminals in t to variables in s.

For example, the head-dependents rule shown in Figure 1 (c) can be formalized as:

t = ((x_1:世界杯) (x_2:在) ($\underline{x_3}$:AD) 举行)
s = x_1 was held x_3 x_2
ϕ = {x_1:世界杯 $\leftrightarrow$ x_1, x_2:在 $\leftrightarrow$ x_2, x_3:AD $\leftrightarrow$ x_3}

where the underline indicates a leaf node, and x_i:*letters* indicates a pair of variable and its constraint.

A derivation is informally defined as a sequence of steps converting a source dependency structure into a target language string, with each step applying one translation rule. As an example, Figure 2 shows the derivation for translating a Chinese (CH) sentence into an English (EN) string.

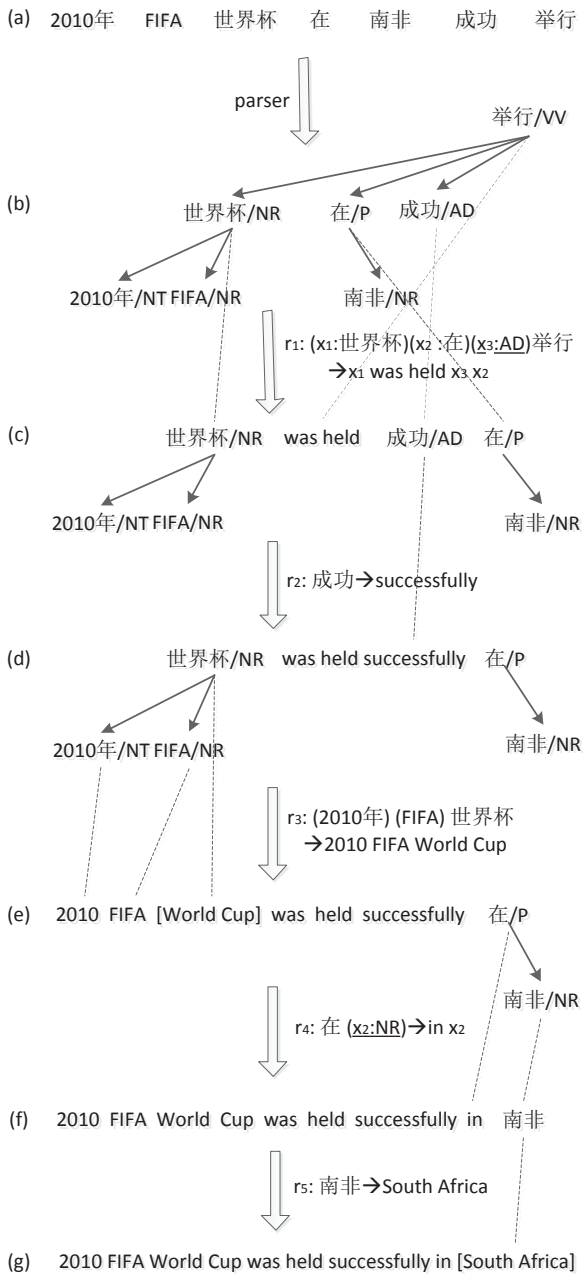

Figure 2: An example derivation of dependency-to-string translation. The dash lines indicate the reordering when employing a head-dependents rule.

CH 2010年 FIFA 世界杯 在 南非 成功 举行

EN *2010 FIFA World Cup was held successfully in South Africa*

The Chinese sentence (a) is first parsed into a dependency structure (b), which is converted into an English string in five steps. First, at the root node, we apply head-dependents rule r_1 shown in Figure 1(c) to translate the top level head-dependents relation and result in three unfinished substructures and target string in (c). The rule is particular interesting since it captures the fact: in Chinese prepositional phrases and adverbs typically modify verbs on the left, whereas in English prepositional phrases and adverbs typically modify verbs on the right. Second, we use head rule r_2 translating "成功" into "successfully" and reach situation (d). Third, we apply head-dependents rule r_3 translating the head-dependents relation rooted at "世界杯" and yield (e). Fourth, head-dependents rules r_5 partially translate the subtree rooted at "在" and arrive situation in (f). Finally, we apply head rule r_5 translating the residual node "南非" and obtain the final translation in (g).

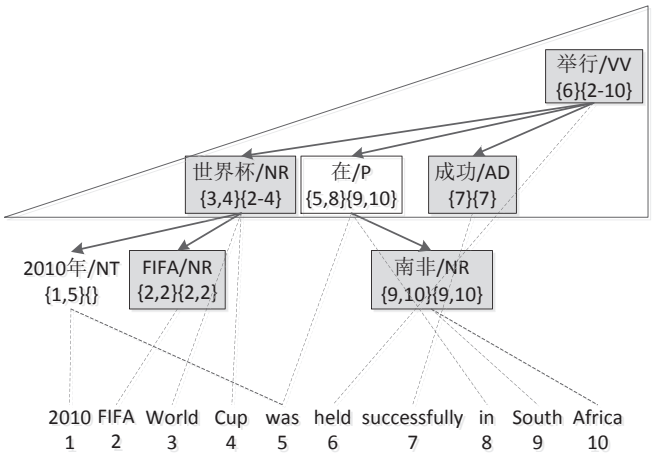

Figure 3: An annotated dependency structure. Each node is annotated with two spans, the former is head span and the latter dependency span. The nodes in *acceptable head set* are displayed in gray, and the nodes in *acceptable dependent set* are denoted by boxes. The triangle denotes the only acceptable head-dependents fragment.

4 Rule Acquisition

The rule acquisition begins with a word-aligned corpus: a set of triples $\langle T, S, A\rangle$, where T is a source dependency structure, S is a target side sentence, and A is an alignment relation between T and S. We extract from each triple $\langle T, S, A\rangle$ head rules that are consistent with the word alignments and head-dependents rules that satisfy the intuition that syntactically close items tend to stay close across languages. We accomplish the rule acquisition through three steps: tree annotation, head-dependents fragments identification and rule induction.

4.1 Tree Annotation

Given a triple $\langle T, S, A\rangle$ as shown in Figure 3, we first annotate each node n of T with two attributes: head span and dependency span, which are defined as follows.

Definition 1. *Given a node n, its **head span** $hsp(n)$ is a set of index of the target words aligned to n.*

For example, hsp(2010年)=$\{1, 5\}$, which corresponds to the target words "*2010*" and "*was*".

Definition 2. *A head span $hsp(n)$ is **consistent** if it satisfies the following property:*

$$\forall_{n' \neq n} hsp(n') \cap hsp(n) = \varnothing.$$

For example, hsp(南非) is consistent, while hsp(2010年) is not consistent since hsp(2010年) $\cap$ hsp(在) $= 5$.

Definition 3. *Given a head span $hsp(n)$, its **closure** cloz(hsp(n)) is the smallest contiguous head span that is a superset of $hsp(n)$.*

For example, $cloz(hsp$(2010年)) = $\{1, 2, 3, 4, 5\}$, which corresponds to the target side word sequence "*2010 FIFA World Cup was*". For simplicity, we use $\{$1-5$\}$ to denotes the contiguous span $\{1, 2, 3, 4, 5\}$.

Definition 4. *Given a subtree T' rooted at n, the **dependency span** $dsp(n)$ of n is defined as:*

$$dsp(n) = cloz(\bigcup_{\substack{n' \in T' \\ hsp(n') \text{ is consistent}}} hsp(n')).$$

If the head spans of all the nodes of T' is not consistent, $dsp(n) = \varnothing$.

For example, since hsp(在) is not consistent, dsp(在)=dsp(南非)=$\{9, 10\}$, which corresponds to the target words "*South*" and "*Africa*".

The tree annotation can be accomplished by a single postorder transversal of T. The extraction of *head rules* from each node can be readily achieved with the same criteria as (Och and Ney, 2004). In

the following, we focus on *head-dependents rules* acquisition.

4.2 Head-Dependents Fragments Identification

We then identify the head-dependents fragments that are suitable for rule induction from the annotated dependency structure.

To facilitate the identification process, we first define two sets of dependency structure related to head spans and dependency spans.

Definition 5. *A **acceptable head set** ahs(T) of a dependency structure T is a set of nodes, each of which has a consistent head span.*

For example, the elements of the acceptable head set of the dependency structure in Figure 3 are displayed in gray.

Definition 6. *A **acceptable dependent set** adt(T) of a dependency structure T is a set of nodes, each of which satisfies:* $dep(n) \neq \varnothing$.

For example, the elements of the acceptable dependent set of the dependency structure in Figure 3 are denoted by boxes.

Definition 7. *We say a head-dependents fragments is **acceptable** if it satisfies the following properties:*

1. *the root falls into acceptable head set;*
2. *all the sinks fall into acceptable dependent set.*

An acceptable head-dependents fragment holds the property that the head span of the root and the dependency spans of the sinks do not overlap with each other, which enables us to determine the reordering in the target side.

The identification of *acceptable* head-dependents fragments can be achieved by a single preorder transversal of the annotated dependency structure. For each accessed internal node n, we check whether the head-dependents fragment f rooted at n is acceptable. If f is acceptable, we output an acceptable head-dependents fragment; otherwise we access the next node.

Typically, each acceptable head-dependents fragment has three types of nodes: *internal nodes*, internal nodes of the dependency structure; *leaf nodes*, leaf nodes of the dependency structure; *head node*, a special internal node acting as the head of the related head-dependents relation.

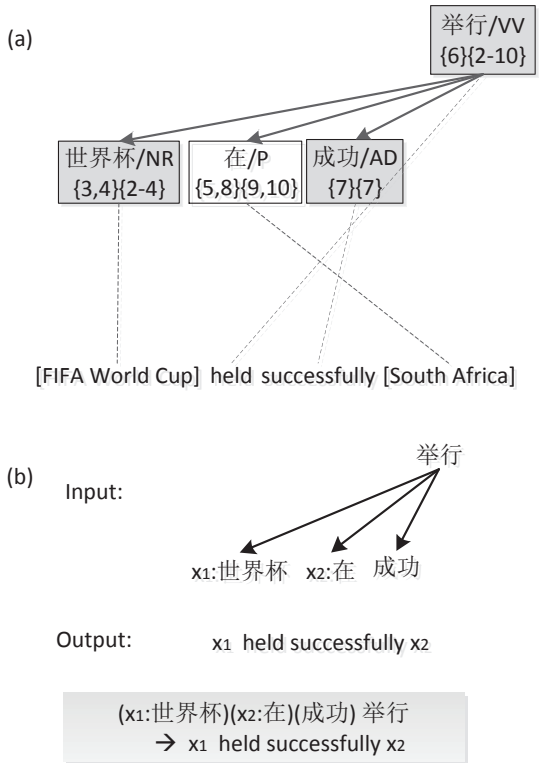

Figure 4: A lexicalized head-dependents rule (b) induced from the only acceptable head-dependents fragment (a) of Figure 3.

4.3 Rule Induction

From each acceptable head-dependents fragment, we induce a set of lexicalized and unlexicalized head-dependents rules.

4.3.1 Lexicalized Rule

We induce a lexicalized head-dependents rule from an acceptable head-dependents fragment by the following procedure:

1. extract the head-dependents relation and mark the internal nodes as substitution sites. This forms the input of a head-dependents rule;
2. place the nodes in order according to the head span of the root and the dependency spans of the sinks, then replace the internal nodes with variables and the other nodes with the target words covered by their head spans. This forms the output of a head-dependents rule.

Figure 4 shows an acceptable head-dependents fragment and a lexicalized head-dependents rule in-

duced from it.

4.3.2 Unlexicalized Rules

Since head-dependents relations with verbs as heads typically consist of more than four nodes, employing only lexicalized head-dependents rules will result in severe sparseness problem. To alleviate this problem, we generalize the lexicalized head-dependents rules and induce rules with unlexicalized nodes.

As we know, the modification relation of a head-dependents relation is determined by the edges. Therefore, we can replace the lexical word of each node with its categories (i.e. POS) and obtain new head-dependents relations with unlexicalized nodes holding the same modification relation. Here we call the lexicalized and unlexicalized head-dependents relations as instances of the modification relation. For a head-dependents relation with m node, we can produce $2^m - 1$ instances with unlexicalized nodes. Each instance represents the modification relation with a different specification.

Based on this observation, from each lexicalized head-dependent rule, we generate new head-dependents rules with unlexicalized nodes according to the following principles:

1. change the aligned part of the target string into a new variable when turning a head node or a leaf node into its category;

2. keep the target side unchanged when turning a internal node into its category.

Restrictions: Since head-dependents relations with verbs as heads typically consists of more than four nodes, enumerating all the instances will result in a massive grammar with too many kinds of rules and inflexibility in decoding. To alleviate these problems, we filter the grammar with the following principles:

1. nodes of the same type turn into their categories simultaneously.

2. as for leaf nodes, only those with open class words can be turned into their categories. In our experiments of this paper, we only turn those dependents with POS tag in the set of {CD,DT,OD,JJ,NN,NR,NT,AD,FW,PN} into their categories.

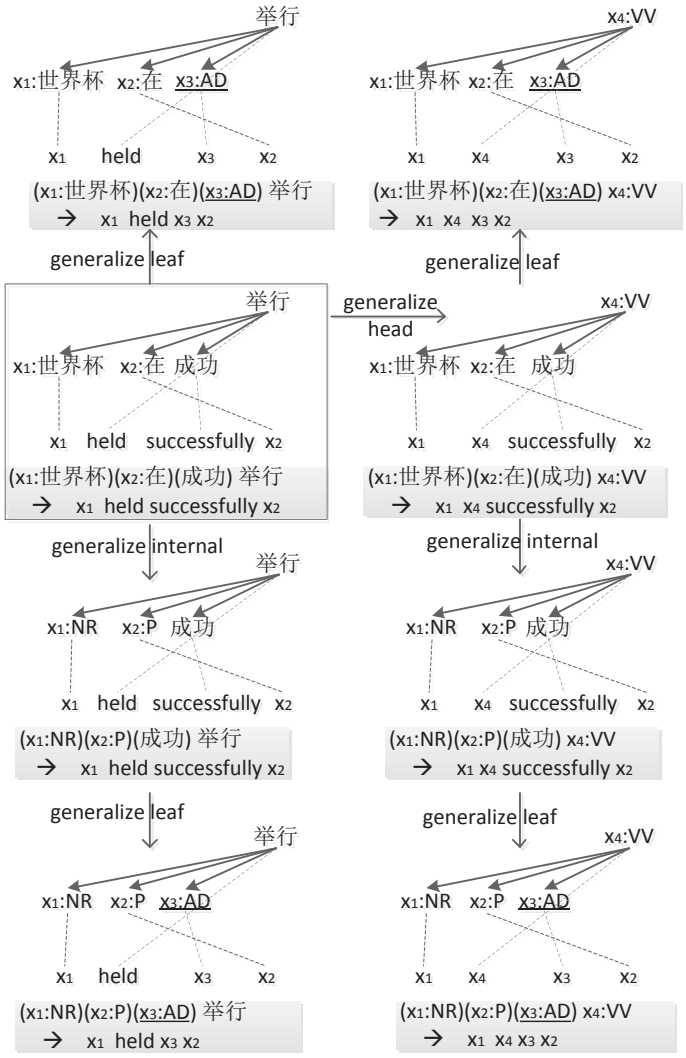

Figure 5: An illustration of rule generalization. Where "x_1:世界杯" and "x_2:在" indicate substitution sites which can be replaced by a subtree rooted at "世界杯" and "在" respectively. "x_3:AD"indicates a substitution site that can be replaced by a subtree whose root has part-of-speech "AD". The underline denotes a leaf node. The box indicates the starting lexicalized head-dependents rule.

Figure 5 illustrates the rule generalization process under these restrictions.

4.3.3 Unaligned Words

We handle the unaligned words of the target side by extending the head spans of the lexicalized head and leaf nodes on both left and right directions. This procedure is similar with the method of (Och and Ney, 2004) except that we might extend several

Algorithm 1: Algorithm for Rule Acquisition

Input: Source dependency structure T, target string S, alignment A
Output: Translation rule set R

```
HSet ← ACCEPTABLE_HEAD(T,S,A)
DSet ← ACCEPTABLE_DEPENDENT(T,S,A)
for each node n ∈ HSet do
    extract head rules
    append the extracted rules to R
    if ∀n' ∈ child(n) n' ∈ DSet
    then
        obtain a head-dependent fragment f
        induce lexicalized and unlexicalized head-dependents rules from f
        append the induced rules to R
    end
end
```

spans simultaneously. In this process, we might obtain $m(m \geq 1)$ head-dependents rules from a head-dependent fragment in handling unaligned words. Each of these rules is assigned with a fractional count $1/m$.

4.4 Algorithm for Rule Acquisition

The rule acquisition is a three-step process, which is summarized in Algorithm 1.

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 $P(s|t)$.

5 The model

Following (Och and Ney, 2002), we adopt a general log-linear model. Let d be a derivation that convert a source dependency structure T 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, we used seven features as follows:

- translation probabilities $P(t|s)$ and $P(s|t)$;
- lexical translation probabilities $P_{lex}(t|s)$ and $P_{lex}(s|t)$;
- rule penalty $exp(-1)$;
- language model $P_{lm}(e)$;
- word penalty $exp(|e|)$.

6 Decoding

Our decoder is based on bottom up chart parsing. It finds the best derivation d^* that convert the input dependency structure into a target string among all possible derivations D:

$$d^* = argmax_{d \in D} P(D) \quad (2)$$

Given a source dependency structure T, the decoder transverses T in post-order. For each accessed internal node n, it enumerates all instances of the related modification relation of the head-dependents relation rooted at n, and checks the rule set for matched translation rules. If there is no matched rule, we construct a *pseudo translation rule* according to the word order of the head-dependents relation. For example, suppose that we can not find any translation rule about to "(2010年) (FIFA) 世界杯", we will construct a pseudo translation rule "(x_1:2010年) (x_2:FIFA) x_3:世界杯 $\rightarrow$ x_1 x_2 x_3". A larger translation is generated by substituting the variables in the target side of a translation rule with the translations of the corresponding dependents. We make use of cube pruning (Chiang, 2007; Huang and Chiang, 2007) to find the k-best items with integrated language model for each node.

To balance performance and speed, we prune the search space in several ways. First, beam thresh-

old β , items with a score worse than β times of the best score in the same cell will be discarded; second, beam size b, items with a score worse than the bth best item in the same cell will be discarded. The item consist of the necessary information used in decoding. Each cell contains all the items standing for the subtree rooted at it. For our experiments, we set $\beta = 10^{-3}$ and $b = 300$. Additionally, we also prune rules that have the same source side ($b = 100$).

7 Experiments

We evaluated the performance of our dependency-to-string model by comparison with replications of the hierarchical phrase-based model and the tree-to-string models on Chinese-English translation.

7.1 Data preparation

Our training corpus consists of 1.5M sentence pairs from LDC data, including LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

We parse the source sentences with Stanford Parser (Klein and Manning, 2003) into projective dependency structures, whose nodes are annotated by POS tags and edges by typed dependencies. In our implementation of this paper, we make use of the POS tags only.

We obtain the word alignments by running GIZA++ (Och and Ney, 2003) on the corpus in both directions and applying "grow-diag-and" refinement(Koehn et al., 2003).

We apply SRI Language Modeling Toolkit (Stolcke, 2002) to train a 4-gram language model with modified Kneser-Ney smoothing on the Xinhua portion of the Gigaword corpus.

We use NIST MT Evaluation test set 2002 as our development set, NIST MT Evaluation test set 2004 (MT04) and 2005 (MT05) as our test set. The quality of translations is evaluated by the *case insensitive* NIST BLEU-4 metric (Papineni et al., 2002).[1]

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.

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

System	Rule #	MT04(%)	MT05(%)
cons2str	30M	34.55	31.94
hiero-re	148M	35.29	33.22
dep2str	56M	**35.82**$^{+}$	**33.62**$^{+}$

Table 1: Statistics of the extracted rules on training corpus and the BLEU scores on the test sets. Where "+" means *dep2str* significantly better than *cons2str* with $p < 0.01$.

7.2 The baseline models

We take a replication of Hiero (Chiang, 2007) as the hierarchical phrase-based model baseline. In our experiments of this paper, we set the beam size $b = 200$ and the beam threshold $\beta = 0$. The maximum initial phrase length is 10.

We use constituency-to-string model (Liu et al., 2006) as the syntax-based model baseline which make use of composed rules (Galley et al., 2006) without handling the unaligned words. In our experiments of this paper, we set the tatTable-limit=20, tatTable-threshold=10^{-1}, stack-limit=100, stack-threshold=10^{-1},hight-limit=3, and length-limit=7.

7.3 Results

We display the results of our experiments in Table 1. Our dependency-to-string model *dep2str* significantly outperforms its constituency structure-based counterpart (*cons2str*) with +1.27 and +1.68 BLEU on MT04 and MT05 respectively. Moreover, without resort to phrases or parse forest, *dep2str* surpasses the hierarchical phrase-based model (*hiero-re*) over +0.53 and +0.4 BLEU on MT04 and MT05 respectively on the basis of a 62% smaller rule set.

Furthermore, We compare some actual translations generated by *cons2str*, *hiero-re* and *dep2str*. Figure 6 shows two translations of our test sets MT04 and MT05, which are selected because each holds a long distance dependency commonly used in Chinese.

In the first example, the Chinese input holds a complex long distance dependencies "巴 尼 耶 在... 与...后 表 示". This dependency corresponds to sentence pattern "noun+prepostional phrase+prepositional phrase+verb", where the former prepositional phrase specifies the position and the latter specifies the time. Both *cons2str* and *hiero-re* are confused by this sentence and mistak-

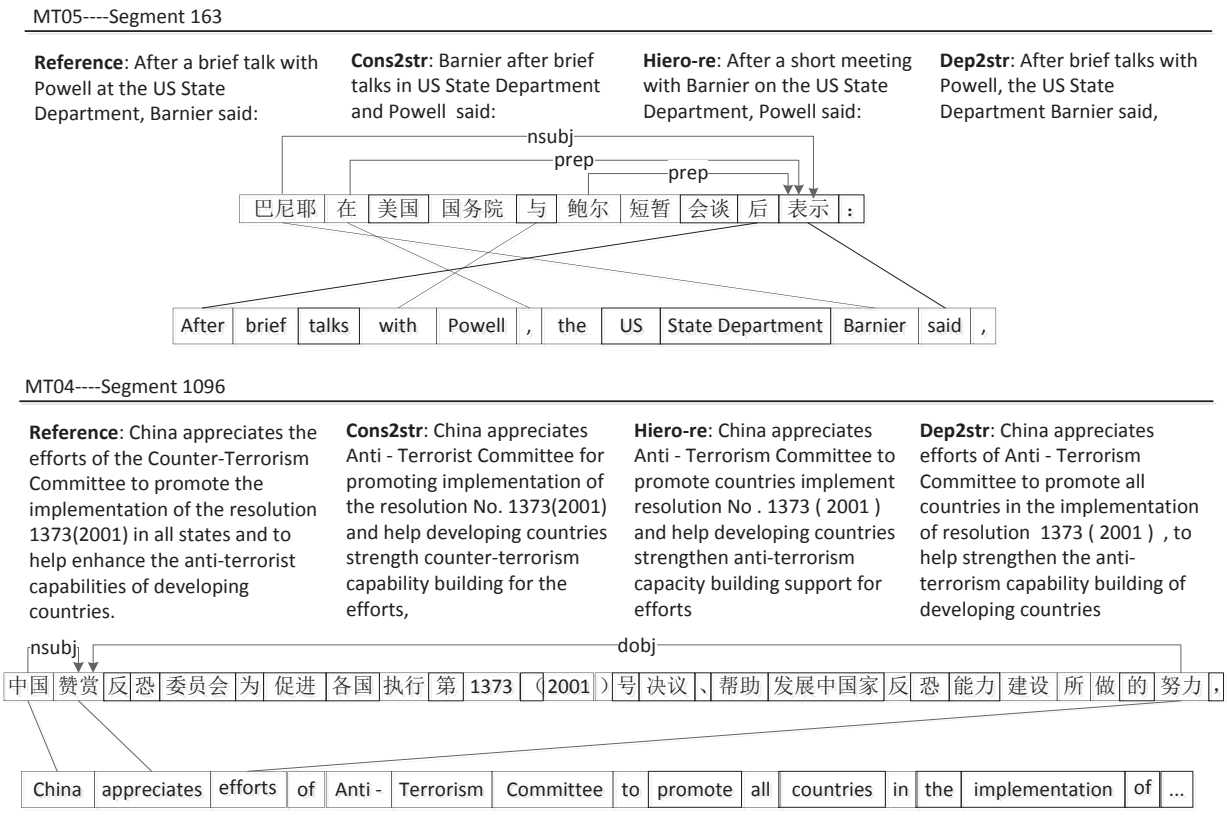

Figure 6: Actual translations produced by the baselines and our system. For our system, we also display the long distance dependencies correspondence in Chinese and English. Here we omit the edges irrelevant to the long distance dependencies.

enly treat "鲍尔(Powell)" as the subjective, thus result in translations with different meaning from the source sentence. Conversely, although "在" is falsely translated into a comma, *dep2str* captures this complex dependency and translates it into "After ... ,(should be *at*) Barnier said", which accords with the reordering of the reference.

In the second example, the Chinese input holds a long distance dependency "中国 赞赏 ... 努力" which corresponds to a simple pattern "noun phrase+verb+noun phrase". However, due to the modifiers of "努力" which contains two sub-sentences including 24 words, the sentence looks rather complicated. *Cons2str* and *hiero-re* fail to capture this long distance dependency and provide monotonic translations which do not reflect the meaning of the source sentence. In contrast, *dep2str* successfully captures this long distance dependency and translates it into "China appreciates efforts of ...", which is almost the same with the reference "China appreciates the efforts of ...".

All these results prove the effectiveness of our dependency-to-string model in both translation and long distance reordering. We believe that the advantage of *dep2str* comes from the characteristics of dependency structures tending to bring semantically related elements together (e.g., verbs become adjacent to all their arguments) and are better suited to lexicalized models (Quirk et al., 2005). And the incapability of *cons2str* and *hiero-re* in handling long distance reordering of these sentences does not lie in the representation of translation rules but the compromises in rule extraction or decoding so as to balance the speed or grammar size and performance. The hierarchical phrase-based model prohibits any nonterminal X from spanning a substring longer than 10 on the source side to make the decoding algorithm asymptotically linear-time (Chiang, 2005).

While constituency structure-based models typically constrain the number of internal nodes (Galley et al., 2006) and/or the height (Liu et al., 2006) of translation rules so as to balance the grammar size and performance. Both strategies limit the ability of the models in processing long distance reordering of sentences with long and complex modification relations.

8 Related Works

As a first step towards semantics, dependency structures are attractive to machine translation. And many efforts have been made to incorporating this desirable knowledge into machine translation.

(Lin, 2004; Quirk et al., 2005; Ding and Palmer, 2005; Xiong et al., 2007) make use of source dependency structures. (Lin, 2004) employs linear paths as phrases and view translation as minimal path covering. (Quirk et al., 2005) extends paths to treelets, arbitrary connected subgraphs of dependency structures, and propose a model based on treelet pairs. Both models require projection of the source dependency structure to the target side via word alignment, and thus can not handle non-isomorphism between languages. To alleviate this problem, (Xiong et al., 2007) presents a dependency treelet string correspondence model which directly map a dependency structure to a target string. (Ding and Palmer, 2005) presents a translation model based on Synchronous Dependency Insertion Grammar(SDIG), which handles some of the non-isomorphism but requires both source and target dependency structures. Most important, all these works do not specify the ordering information directly in translation rules, and resort to either heuristics (Lin, 2004; Xiong et al., 2007) or separate ordering models(Quirk et al., 2005; Ding and Palmer, 2005) to control the word order of translations. By comparison, our model requires only source dependency structure, and handles non-isomorphism and ordering problems simultaneously by directly specifying the ordering information in the head-dependents rules that represent the source side as head-dependents relations and the target side as strings.

(Shen et al., 2008) exploits target dependency structures as dependency language models to ensure the grammaticality of the target string. (Shen et al., 2008) extends the hierarchical phrase-based model and present a string-to-dependency model, which employs string-to-dependency rules whose source side are string and the target as well-formed dependency structures. In contrast, our model exploits source dependency structures, as a tree-based system, it run much faster (linear time vs. cubic time, see (Huang et al., 2006)).

9 Conclusions and future work

In this paper, we present a novel dependency-to-string model, which employs head-dependents rules that represent the source side as head-dependents relations and the target side as string. The head-dependents rules specify the ordering information directly and require only substitution operation. Thus, our model does not need heuristics or ordering model of the previous works to control the word order of translations. Large scale experiments show that our model exhibits good performance in long distance reordering and outperforms the state-of-the-art constituency-to-string model and hierarchical phrase-based model without resort to phrases and parse forest. For the first time, a source dependency-based model shows improvement over the state-of-the-art translation models.

In our future works, we will exploit the semantic information encoded in the dependency structures which is expected to further improve the translations, and replace 1-best dependency structures with dependency forests so as to alleviate the influence caused by parse errors.

Acknowledgments

This work was supported by National Natural Science Foundation of China, Contract 60736014, 60873167, 90920004. We are grateful to the anonymous reviewers for their thorough reviewing and valuable suggestions. We appreciate Yajuan Lv, Wenbin Jiang, Hao Xiong, Yang Liu, Xinyan Xiao, Tian Xia and Yun Huang for the insightful advices in both experiments and writing. Special thanks goes to Qian Chen for supporting my pursuit all through.

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of*

ACL 2005, pages 263–270.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of ACL 2005*.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *In Proceedings of EMNLP 2002*, pages 304–311.

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 ACL 2006*, pages 961–968, Sydney, Australia, July. Association for Computational Linguistics.

Peter Hellwig. 2006. Parsing with dependency grammars. In *Dependenz und Valenz / Dependency and Valency*, volume 2, pages 1081–1109. Berlin, New York.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of ACL 2007*, pages 144–151, Prague, Czech Republic, June.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. A syntax-directed translator with extended domain of locality. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 1–8, New York City, New York, June. Association for Computational Linguistics.

Richard Hudson. 1990. *English Word Grammar*. Blackell.

Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS*, pages 3–10. MIT Press.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Canada, July.

Dekang Lin. 2004. A path-based transfer model for machine translation. In *Proceedings of Coling 2004*, pages 625–630, Geneva, Switzerland, Aug 23–Aug 27.

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

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

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.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL-2003*, pages 160–167, Sapporo, Japan, July.

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

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of ACL 2005*, 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 2008: HLT*, pages 577–585, Columbus, Ohio, June. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of ICSLP, volume 30*, pages 901–904.

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, Prague, Czech Republic, June.

Arnold M. Zwicky. 1985. Heads. *Journal of Linguistics*, 21:1–29.