

# Forced Derivation Tree based Model Training to Statistical Machine Translation

Nan Duan

Microsoft Research Asia

Mu Li

Microsoft Research Asia

Ming Zhou

Microsoft Research Asia

nanduan@microsoft.com muli@microsoft.com mingzhou@microsoft.com

## Abstract

A *forced derivation tree (FDT)* of a sentence pair  $\{f, e\}$  denotes a derivation tree that can translate  $f$  into its accurate target translation  $e$ . In this paper, we present an approach that leverages structured knowledge contained in FDTs to train component models for statistical machine translation (SMT) systems. We first describe how to generate different FDTs for each sentence pair in training corpus, and then present how to infer the optimal FDTs based on their derivation and alignment qualities. As the first step in this line of research, we verify the effectiveness of our approach in a BTG-based phrasal system, and propose four FDT-based component models. Experiments are carried out on large scale English-to-Japanese and Chinese-to-English translation tasks, and significant improvements are reported on both translation quality and alignment quality.

## 1 Introduction

Most of today's SMT systems depends heavily on parallel corpora aligned at the word-level to train their different component models. However, such annotations do have their drawbacks in training.

On one hand, word links predicted by automatic aligners such as GIZA++ (Och and Ney, 2004) often contain errors. This problem gets even worse on language pairs that differ substantially in word orders, such as English and Japanese/Korean/German. The descent of the word alignment quality will lead to inaccurate component models straightforwardly.

On the other hand, several component models are designed to supervise the decoding procedures,

which usually rely on training examples extracted from word-aligned sentence pairs, such as distortion models (Tillman, 2004; Xiong et al., 2006; Galley and Manning, 2008) and sequence models (Banchs et al., 2005; Quirk and Menezes, 2006; Vaswani et al., 2011). Ideally, training examples of models are expected to match most of the situations that could be met in decoding procedures. But actually, plain structures of word alignments are too coarse to provide enough knowledge to ensure this expectation.

This paper presents an FDT-based model training approach to SMT systems by leveraging structured knowledge contained in FDTs. An FDT of a sentence pair  $\{f, e\}$  denotes a derivation tree that can translate  $f$  into its accurate target translation  $e$ . The principle advantage of this work is two-fold. First, using alignments induced from the 1-best FDTs of all sentence pairs, the overall alignment quality of training corpus can be improved. Second, comparing to word alignments, FDTs can provide richer structured knowledge for various component models to extract training instances. Our FDT-based model training approach performs via three steps: (1) **generation**, where an *FDT space* composed of different FDTs is generated for each sentence pair in training corpus by the *forced decoding* technique; (2) **inference**, where the optimal FDTs are extracted from the FDT space of each sentence pair based on both derivation and alignment qualities measured by a *memory-based re-ranking model*; (3) **training**, where various component models are trained based on the optimal FDTs extracted in the inference step.

Our FDT-based model training approach can be adapted to SMT systems with arbitrary paradigms.

As the first step in this line of research, our approach is verified in a phrase-based SMT system on both English-to-Japanese and Chinese-to-English translation tasks. Significant improvements are reported on both translation quality (up to 1.31 BLEU) and word alignment quality (up to 3.15 F-score).

## 2 Forced Derivation Tree for SMT

A forced derivation tree (**FDT**) of a sentence pair  $\{f, e\}$  can be defined as a pair  $\mathcal{G} = \langle \mathcal{D}, \mathcal{A} \rangle$ :

- $\mathcal{D}$  denotes a derivation that can translate  $f$  into  $e$  accurately, using a set of translation rules.
- $\mathcal{A}$  denotes a set of word links  $(i, j)$  indicating that  $e_i \in e$  aligns to  $f_j \in f$ .

In this section, we first describe how to generate FDTs for each sentence pair in training corpus, which is denoted as the *generation* step, and then present how to select the optimal FDT for each sentence pair, which is denoted as the *inference* step. We leave a real application of FDTs to the model training in a phrase-based SMT system in Section 3.

### 2.1 Generation

We first describe how to generate multiple FDTs for each sentence pair in training corpus  $\mathcal{C}$  based on the forced decoding (**FD**) technique, which performs via the following four steps:

1. Train component models needed for a specific SMT paradigm  $\mathcal{M}$  based on training corpus  $\mathcal{C}$ ;
2. Perform MERT on the development data set to obtain a set of optimized feature weights;
3. For each  $\{f, e\} \in \mathcal{C}$ , translate  $f$  into accurate  $e$  based on  $\mathcal{M}$ , component models trained in step 1, and feature weights optimized in step 2;
4. For each  $\{f, e\} \in \mathcal{C}$ , output the hypergraph (Huang and Chiang, 2005)  $\mathcal{H}(f, e)$  generated in step 3 as its *FDT space*.

In step 3: (1) all partial hypotheses that do not match any sequence in  $e$  will be discarded; (2) derivations covering identical source and target words but with different alignments will be kept as different partial candidates, as they can produce different FDTs for

the same sentence pair. For each  $\{f, e\}$ , the probability of each  $\mathcal{G} \in \mathcal{H}(f, e)$  is computed as:

$$p(\mathcal{G}|\mathcal{H}(f, e)) = \frac{\exp\{\psi(\mathcal{G})\}}{\sum_{\mathcal{G}' \in \mathcal{H}(f, e)} \exp\{\psi(\mathcal{G}')\}} \quad (1)$$

where  $\psi(\mathcal{G})$  is the FD model score assigned to  $\mathcal{G}$ .

For each sentence pair, different alignment candidates can be induced from its different forced derivation trees generated in the generation step, because FD can use phrase pairs with different internal word links extracted from other sentence pairs to reconstruct the given sentence pair, which could lead to better word alignment candidates.

### 2.2 Inference

Given an FDT space  $\mathcal{H}(f, e)$ , we propose a memory-based re-ranking model (**MRM**), which selects the best FDT  $\hat{\mathcal{G}}$  as follows:

$$\begin{aligned} \hat{\mathcal{G}} &= \underset{\mathcal{G} \in \mathcal{H}(f, e)}{\operatorname{argmax}} \frac{\exp\{\sum_i \lambda_i h_i(\mathcal{G})\}}{\sum_{\mathcal{G}' \in \mathcal{H}(f, e)} \exp\{\sum_i \lambda_i h_i(\mathcal{G}')\}} \\ &= \underset{\mathcal{G} \in \mathcal{H}(f, e)}{\operatorname{argmax}} \sum_i \lambda_i h_i(\mathcal{G}) \end{aligned} \quad (2)$$

where  $h_i(\mathcal{G})$  is feature function and  $\lambda_i$  is its feature weight. Here, *memory* means the whole translation history that happened in the generation step will be used as the evidence to help us compute features.

From the definition we can see that the quality of an FDT directly relates to two aspects: its derivation  $\mathcal{D}$  and alignments  $\mathcal{A}$ . So two kinds of features are used to measure the overall quality of each FDT.

(I) The features in the first category measure the derivation quality of each FDT, including:

- $h(\bar{e}|\bar{f})$ , source-to-target translation probability of a translation rule  $r = \{\bar{f}, \bar{e}\}$ .

$$h(\bar{e}|\bar{f}) = \frac{\sum_{\{f, e\} \in \mathcal{C}} \operatorname{frac}_{\mathcal{H}(f, e)}(\bar{f}, \bar{e})}{\sum_{\{f, e\} \in \mathcal{C}} \sum_{e'} \operatorname{frac}_{\mathcal{H}(f, e)}(\bar{f}, e')} \quad (3)$$

$\operatorname{frac}_{\mathcal{H}(f, e)}(\bar{f}, \bar{e})$  denotes the fractional count of  $r$  used in generating  $\mathcal{H}(f, e)$ :

$$\operatorname{frac}_{\mathcal{H}(f, e)}(\bar{f}, \bar{e}) = \sum_{\mathcal{G} \in \mathcal{H}(f, e)} 1_r(\mathcal{G}) p(\mathcal{G}|\mathcal{H}(f, e))$$

$1_r(\mathcal{G})$  is an indicator function that equals 1 when  $r$  is used in  $\mathcal{G}$  and 0 otherwise. In practice, we use  $p_{\mathcal{H}(f, e)}(r)$  of  $r$  to approximate

$frac_{\mathcal{H}(f,e)}(\bar{f}, \bar{e})$  when the size of  $\mathcal{H}(f, e)$  is too large to enumerate all FDTs:

$$p_{\mathcal{H}(f,e)}(r) = \frac{\omega(r)\mathcal{O}(\text{head}(r))\prod_{v\in\text{tail}(r)}\mathcal{I}(v)}{\mathcal{Z}(f)}$$

where  $\omega(r)$  is the weight of translation rule  $r$  in the FDT space  $\mathcal{H}(f, e)$ ,  $\mathcal{Z}$  is a normalization factor that equals to the inside probability of the root node in  $\mathcal{H}(f, e)$ ,  $\mathcal{I}(v)$  and  $\mathcal{O}(v)$  are the standard inside and outside probabilities of a node  $v$  in  $\mathcal{H}(f, e)$ ,  $\text{head}(r)$  and  $\text{tail}(r)$  are the head node and a set of tail nodes of a translation rule  $r$  in  $\mathcal{H}(f, e)$  respectively.

- $h(\bar{f}|\bar{e})$ , target-to-source translation probability of a translation rule  $r = \{\bar{f}, \bar{e}\}$ .

$$h(\bar{f}|\bar{e}) = \frac{\sum_{\{f,e\}\in\mathcal{C}}frac_{\mathcal{H}(f,e)}(\bar{f}, \bar{e})}{\sum_{\{f,e\}\in\mathcal{C}}\sum_{\bar{f}'}frac_{\mathcal{H}(f,e)}(\bar{f}', \bar{e})} \quad (4)$$

- $h_{\#}(r)$ , smoothed usage count for translation rule  $r = \{\bar{f}, \bar{e}\}$  in the whole generation step.

$$h_{\#}(r) = \frac{1}{1 + e^{\{-\sum_{\{f,e\}\in\mathcal{C}}frac_{\mathcal{H}(f,e)}(\bar{f}, \bar{e})\}}} \quad (5)$$

In this paper, the sigmoid function is used to make sure that the feature values of different translation rules are in a proper value range.

- $h_r(\mathcal{G})$ , number of translation rules used in  $\mathcal{G}$ .
- $h_d(\mathcal{G})$ , structure-based score of  $\mathcal{G}$ . For FDTs generated by phrase-based paradigms, it can be computed by distortion models; while for FDTs generated by syntax-based paradigms, it can be computed by either parsing models or syntactic LMs (Charniak et al., 2003).

The overfitting issue in the generation step can be alleviated by leveraging memory-based features in the inference step.  $h_{\#}(r)$  is used to penalize those long translation rules which tend to occur in only a few training sentences and are used few times in FD,  $h_r(\mathcal{G})$  adjust our MRM to prefer FDTs consisting of more translation rules,  $h_d(\mathcal{G})$  is used to select FDTs with better parse tree-like structures, which can be induced from their derivations directly.

(II) The features in the second category measure the alignment quality of each FDT, including:

- word pair translation probabilities trained from IBM models (Brown et al., 1993);
- log-likelihood ratio (Moore, 2005);
- conditional link probability (Moore, 2005);
- count of unlinked words;
- counts of inversion and concatenation.

Many alignment-inspired features can be used in MRM. This paper only uses those commonly-used ones that have already been proved useful in many previous work (Moore, 2005; Moore et al., 2006; Fraser and Marcu, 2006; Liu et al., 2010).

Following the common practice in SMT research, the MERT algorithm (Och, 2003) is used to tune feature weights in MRM. Due to the fact that all FDTs of each sentence pair share identical translation, we cannot use BLEU as the error criterion any more. Instead, alignment F-score is used as the alternative. We will show in Section 5 that after the inference step, alignment quality can be improved by replacing original alignments of each sentence pair with alignments induced from its 1-best FDT. Future work could experiment with other error criterions, such as reordering-based loss functions (Birch et al., 2010; Talbot et al., 2011; Birch and Osborne, 2011) or span F1 (DeNero and Uszkoreit, 2011).

### 3 Training in Phrase-based SMT

As the first step in this line of research, we explore the usage of FDT-based model training method in a phrase-based SMT system (Xiong et al., 2006), which employs Bracketing Transduction Grammar (BTG) (Wu, 1997) to parse parallel sentences. The reason of choosing this system is due to the prominent advantages of BTG, such as the simplicity of the grammar and the good coverage of syntactic diversities between different language pairs. We first describe more details of FDTs under BTG. Then, four FDT-based component models are presented.

#### 3.1 BTG-based FDT

Given a sentence pair  $f = \{f_0, \dots, f_J\}$  and  $e = \{e_0, \dots, e_I\}$  in training corpus, its FDT  $\mathcal{G}$  generated based on BTG is a binary tree, which is presented by a set of terminal translation states  $\mathcal{T}$  and a set of non-terminal translation states  $\mathcal{N}$ , where:

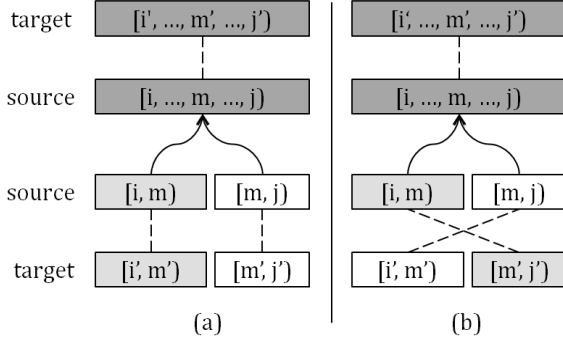


Figure 1:  $\mathcal{S} = \{\bar{f}_{[i,j]}, \bar{e}_{[i',j']}, \bar{\mathcal{A}}, m, m', \mathcal{R}\}$  is denoted by the dark-shaded rectangle pair. It can be split into two child translation states,  $\mathcal{S}_l$ , which is denoted by the light-shaded rectangle pair, and  $\mathcal{S}_r$ , which is denoted by the white rectangle pair. Dash lines within rectangle pairs denote their internal alignments and solid lines with rows denote BTG rules. (a) uses  $[\cdot]$  to combine two translation states, while (b) uses  $\langle \cdot \rangle$ . Both  $\mathcal{S}_l$  and  $\mathcal{S}_r$  belong to  $\mathcal{T} \cup \mathcal{N}$ .

- each *terminal translation state*  $\mathcal{S} \in \mathcal{T}$  is a 3-tuple  $\{\bar{f}_{[i,j]}, \bar{e}_{[i',j']}, \bar{\mathcal{A}}\}$ , in which  $\bar{f}_{[i,j]}$  denotes the word sequence that covers the source span  $[i, j]$  of  $f$ ,  $\bar{e}_{[i',j']}$  denotes the target translation of  $\bar{f}_{[i,j]}$ , which is the word sequence that covers the target span  $[i', j']$  of  $e$  at the same time,  $\bar{\mathcal{A}}$  is a set of word links that aligns  $\bar{f}_{[i,j]}$  and  $\bar{e}_{[i',j']}$ .
- each *non-terminal translation state*  $\mathcal{S} \in \mathcal{N}$  is a 5-tuple  $\{\bar{f}_{[i,j]}, \bar{e}_{[i',j']}, \bar{\mathcal{A}}, m, m', \mathcal{R}\}$ <sup>1</sup>. The first 3 elements have the same meanings as in  $\mathcal{T}$ , while  $m$  and  $m'$  denote two split points that divide  $\mathcal{S}$  into two child translation states,  $\mathcal{S}_l$  and  $\mathcal{S}_r$ ,  $\mathcal{R}$  denotes a BTG rule, which is either a  $[\cdot]$  operation or a  $\langle \cdot \rangle$  operation<sup>2</sup>. The relationship between  $\mathcal{S}_l$ ,  $\mathcal{S}_r$  and  $\mathcal{S}$  is illustrated in Figure 1.

All terminal translation states of the sentence pair  $\{f, e\}$  are disjoint but cover  $f_{[0, J+1]}$  and  $e_{[0, I+1]}$  at the same time, where  $J = |f|$  and  $I = |e|$ , and all non-terminal translation states correspond to the partial decoding states generated during decoding.

### 3.2 FDT-based Translation Model

First, an FDT-based translation model (**FDT-TM**) is presented for our BTG-based system.

<sup>1</sup>We sometimes omit  $m, m'$  and  $\mathcal{R}$  for a simplicity reason.

<sup>2</sup>A  $[\cdot]$  operation combines the translations of two consecutive source spans  $[i, m]$  and  $[m, j]$  in a monotonic way; while a  $\langle \cdot \rangle$  operation combines them in an inverted way.

Given sentence pairs in training corpus with their corresponding FDT spaces, we train FDT-TM in two different ways: (1) The first only uses the 1-best FDT of each sentence pair. Based on each alignment  $\mathcal{A}$  induced from each 1-best FDT  $\mathcal{G}$ , all possible bilingual phrases are extracted. Then, the maximum likelihood estimation (MLE) is used to compute probabilities and generate an FDT-TM. (2) The second uses the  $n$ -best FDTs of each sentence pair, which is motivated by several studies (Venugopal et al., 2008; Liu et al., 2009). For each sentence pair  $\{f, e\}$ , we first induce  $n$  alignments  $\{\mathcal{A}_1, \dots, \mathcal{A}_n\}$  from the top  $n$  FDTs  $\Omega = \{\mathcal{G}_1, \dots, \mathcal{G}_n\} \subset \mathcal{H}(f, e)$ . Each  $\mathcal{A}_k$  is annotated with the posterior probability of its corresponding FDT  $\mathcal{G}_k$  as follows:

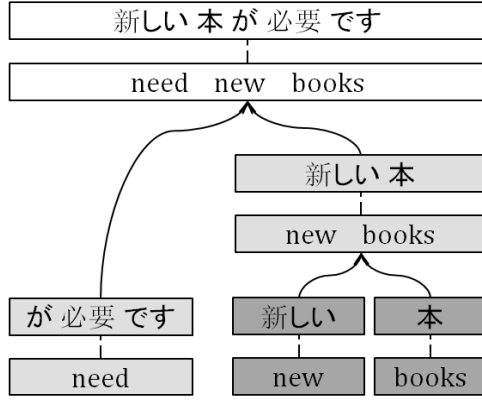
$$p(\mathcal{A}_k | \mathcal{G}_k) = \frac{\exp\{\sum_i \lambda_i h_i(\mathcal{G}_k)\}}{\sum_{\mathcal{G}_{k'} \in \Omega} \exp\{\sum_i \lambda_i h_i(\mathcal{G}_{k'})\}} \quad (6)$$

where  $\sum_i \lambda_i h_i(\mathcal{G}_k)$  is the model score assigned to  $\mathcal{G}_k$  by MRM. Then, all possible bilingual phrases are extracted from the expanded training corpus built using  $n$ -best alignments for each sentence pair. The count of each phrase pair is now computed as the sum of posterior probabilities, instead of the sum of absolute frequencies. Last, MLE is used to compute probabilities and generate an FDT-TM.

### 3.3 FDT-based Distortion Model

In Xiong’s BTG system, training instances of the distortion model (DM) are pruned based on heuristic rules, aiming to keep the training size acceptable. But this will cause the examples remained cannot cover all reordering cases that could be met in real decoding procedures. To overcome this drawback, we propose an FDT-based DM (**FDT-DM**).

Given the 1-best FDT  $\mathcal{G}$  of a sentence pair  $\{f, e\}$ , all non-terminal translation states  $\{\mathcal{S}_1, \dots, \mathcal{S}_K\}$  are first extracted. For each  $\mathcal{S}_k$ , we split it into two child translation states  $\mathcal{S}_{kl}$  and  $\mathcal{S}_{kr}$ . A training instance can be then obtained, using the BTG operation  $\mathcal{R} \in \mathcal{S}_k$  as its class label and boundary words of two translation blocks  $(\bar{f}_{\mathcal{S}_{kl}}, \bar{e}_{\mathcal{S}_{kl}})$  and  $(\bar{f}_{\mathcal{S}_{kr}}, \bar{e}_{\mathcal{S}_{kr}})$  contained in  $\mathcal{S}_{kl}$  and  $\mathcal{S}_{kr}$  as its features. Last, the FDT-DM is trained based on all training instances by a MaxEnt toolkit, which can cover both local and global reordering situations due to its training instance extraction mechanism. Figure 2 shows an example of extracting training instances from an FDT.



$\{0, (\text{new}, \text{新しい}) (\text{books}, \text{本})\}$   
 $\{1, (\text{need}, \text{が 必要 です}) (\text{new books}, \text{新しい 本})\}$

Figure 2: An example of extracting training instances from an FDT, where solid lines with rows denote BTG operations and dash lines denote alignments. Two instances can be extracted from this FDT, where 0 and 1 denote a  $[\cdot]$  operation and a  $\langle \cdot \rangle$  operation respectively. In DM training, the number (0 or 1) in each instance is used as a label, while boundary words are extracted from each instance’s two phrase pairs and used as lexical features.

### 3.4 FDT-based Source Language Model

We next propose an FDT-based source language model (**FDT-SLM**).

Given the 1-best FDT  $\mathcal{G}$  of a sentence pair  $\{f, e\}$ , we first extract a reordered source word sequence  $f' = \{f'_0, \dots, f'_j\}$  from  $\mathcal{G}$ , based on the order of terminal translation states in  $\mathcal{G}$  which covers the target translation  $e$  from left to right. This procedure can be illustrated by Algorithm 1. Then, all reordered source sentences of training corpus are used to train a source LM. During decoding, each time when a new hypothesis is generated, we obtain its reordered source word sequence as well, compute a LM score based on FDT-SLM and use it as a new feature:

$$h_{SLM}(f') = \prod_{k=1}^J p(f'_k | f'_{k-n+1}, \dots, f'_{k-1}) \quad (7)$$

### 3.5 FDT-based Rule Sequence Model

The last contribution in this section is an FDT-based rule sequence model (**FDT-RSM**).

Given the 1-best FDT  $\mathcal{G}$  of a sentence pair  $\{f, e\}$ , we first extract a sequence of translation rule applications  $\{r_1, \dots, r_K\}$  based on Algorithm 2, where

---

#### Algorithm 1: Sequence Extraction in FDT-SLM

---

- 1 let  $f' = \emptyset$ ;
  - 2 let  $\bar{\mathcal{S}} = \{\mathcal{S}_{1'}, \dots, \mathcal{S}_{K'}\}$  represents an ordered sequence of terminal translation states whose target phrases cover  $e$  from left to right orderly;
  - 3 **foreach**  $\mathcal{S} \in \bar{\mathcal{S}}$  in the left-to-right order **do**
  - 4 | extract  $\bar{f}_{[i,j]}$  from  $\mathcal{S}$ ;
  - 5 | append  $\bar{f}_{[i,j]}$  to  $f'$ ;
  - 6 | append a blank space to  $f'$ ;
  - 7 **end**
  - 8 **return**  $f'$  as a reordered source word sequence.
- 

$r_k = (\bar{f}_{[i,j]}, \bar{e}_{[i',j']})$  denotes the  $k^{\text{th}}$  phrase pair. Figure 3 gives an example of extracting a rule sequence from an FDT. An FDT-RSM is trained based on all rule sequences extracted from training corpus. During decoding, each time when a new hypothesis is generated, we compute an FDT-RSM score based on its rule sequence and use it as a new feature:

$$h_{RSM}(f, e) = \prod_{k=1}^K p(r_k | r_{k-n+1}, \dots, r_{k-1}) \quad (8)$$

---

#### Algorithm 2: Sequence Extraction in FDT-RSM

---

- 1 let  $r' = \emptyset$ ;
  - 2 let  $\bar{\mathcal{S}} = \{\mathcal{S}_{1'}, \dots, \mathcal{S}_{K'}\}$  represents an ordered sequence of terminal translation states whose target phrases cover  $e$  from left to right orderly;
  - 3 **foreach**  $\mathcal{S} \in \bar{\mathcal{S}}$  in the left-to-right order **do**
  - 4 | extract a phrase pair  $(\bar{f}_{[i,j]}, \bar{e}_{[i',j']})$  from  $\mathcal{S}$ ;
  - 5 | add  $r_k = (\bar{f}_{[i,j]}, \bar{e}_{[i',j']})$  to  $r'$ ;
  - 6 **end**
  - 7 **return**  $r'$  as a rule sequence.
- 

The main difference between FDT-SLM and FDT-RSM is that the former is trained based on monolingual n-grams; while the latter is trained based on bilingual phrases. Although these two models are trained and computed in an LM style, they are used as reordering features, because they help SMT decoder find better decoding sequences.

Of course, the usage of FDTs need not be limited to the BTG-based system, and we consider using FDTs generated by SCFG-based systems or traditional left-to-right phrase-based systems in future.

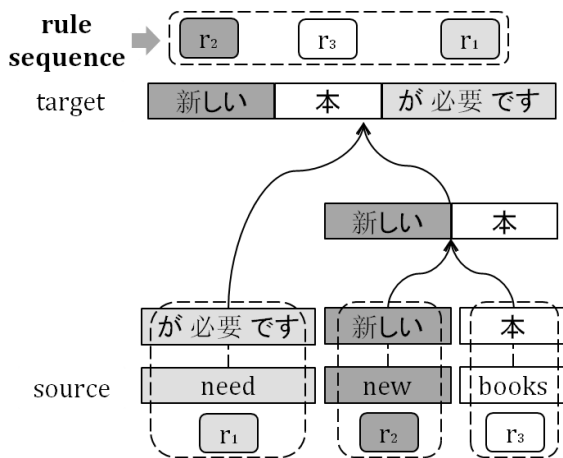


Figure 3: An example of extracting a rule sequence from an FDT. In order to generate the correct target translation, the desired rule sequence should be  $r_2 \Rightarrow r_3 \Rightarrow r_1$ .

## 4 Related Work

### 4.1 Forced Decoding/Alignment

Schwartz (2008) used forced decoding to leverage multilingual corpus to improve translation quality; Shen et al. (2008) used forced alignment to train a better phrase segmentation model; Wuebker et al. (2010) used forced alignment to re-estimate translation probabilities using a leaving-one-out strategy. We consider the usage of FD in Section 2.1 to be a direct extension of these approaches, but one that generates FDTs for parallel data rather than focusing on phrase segmentation or probability estimation.

### 4.2 Pre-reordering

Pre-reordering (PRO) techniques (Collins et al., 2005; Xu et al., 2009; Genzel et al., 2010; Lee et al., 2010) used features from syntactic parse trees to reorder source sentences at training and translation time. A parser is often indispensable to provide syntactic information for such methods. Recently, DeNero and Uszkoreit (2011) proposed an approach that induced parse trees automatically from word-aligned training corpus to perform PRO for a phrase-based SMT system, instead of relying on treebanks. First, binary parse trees are induced from word-aligned training corpus. Based on them, a monolingual parsing model and a tree reordering model are trained to pre-reorder source words into the target-language-like order. Their work is distinct from ours

because it focused on inducing sentence structures for the PRO task, but mirrors ours in demonstrating that there is a potential role for structure-based training corpus in SMT model training.

### 4.3 Distortion Models

Lexicalized distortion models (Tillman, 2004; Zens and Ney, 2006; Xiong et al., 2006; Galley and Manning, 2008;) are widely used in phrase-based SMT systems. Training instances of these models are extracted from word-aligned sentence pairs. Due to efficiency reasons, only parts of all instances are kept and used in DM training, which cannot cover all possible reordering situations that could be met in decoding. In FDT-DM, by contrast, training instances are extracted from FDTs. Such instances take both local and global reordering cases into consideration.

### 4.4 Sequence Models

Feng et al. (2010) proposed an SLM in a phrase-based SMT system. They used it as a reordering feature in the sense that it helped the decoder to find correct decoding sequences. The difference between their model and our FDT-SLM is that, in their work, the reordered source sequences are extracted based on word alignments only; while in our FDT-SLM, such sequences are obtained based on FDTs.

Quirk and Menezes (2006) proposed a Minimal Translation Unit (MTU)-based sequence model and used it in their treelet system; Vaswani et al. (2011) proposed a rule Markov model to capture dependencies between minimal rules for a top-down tree-to-string system. The key difference between FDT-RSM and previous work is that the rule sequences are extracted from FDTs, and no parser is needed.

## 5 Experiments

### 5.1 Data and Metric

Experiments are carried out on English-to-Japanese (E-J) and Chinese-to-English (C-E) MT tasks.

For *E-J task*, bilingual data used contains 13.3M sentence pairs after pre-processing. The Japanese side of bilingual data is used to train a 4-gram LM. The development set (*dev*) which contains 2,000 sentences is used to optimize the log-linear SMT model. Two test sets are used for evaluation, which contain 5,000 sentences (*test-1*) and 999 sentences

(*test-2*) respectively. In all evaluation data sets, each source sentence has only one reference translation.

For *C-E task*, bilingual data used contains 0.5M sentence pairs with high translation quality, including LDC2003E07, LDC2003E14, LDC2005T06, LDC2005T10, LDC2005E83, LDC2006E26, LDC2006E34, LDC2006E85 and LDC2006E92. A 5-gram LM is trained on the Xinhua portion of LDC English Gigaword Version 3.0. NIST 2004 (*MT04*) data set is used as dev set, and evaluation results are measured on NIST 2005 (*MT05*) and NIST 2008 (*MT08*) data sets. In all evaluation data sets, each source sentence has four reference translations.

Default word alignments for both SMT tasks are performed by GIZA++ with the *intersect-diag-grow* refinement. Translation quality is measured in terms of *case-insensitive BLEU* (Papineni et al., 2002) and reported in percentage numbers.

## 5.2 Baseline System

The phrase-based SMT system proposed by Xiong et al. (2006) is used as the baseline system, with a MaxEnt principle-based lexicalized reordering model integrated, which is used to handle reorderings in decoding. The maximum lengths for the source and target phrases are 5 and 7 on E-J task, and 3 and 5 on C-E task. The beam size is set to 20.

## 5.3 Translation Quality on E-J Task

We first evaluate the effectiveness of our FDT-based model training approach on E-J translation task, and present evaluation results in Table 1, in which *BTG* denotes the performance of the baseline system.

*FDT-TM* denotes the improved system that uses FDT-TM proposed in Section 3.2 instead of original phrase table. As described in Section 3.2, we tried different sizes of n-best FDTs to induce alignments for phrase extraction and found the optimal choice is 5. Besides, in order to make full use of the training corpus, for those sentence pairs that are failed in FD, we just use their original word alignments to extract bilingual phrases. We can see from Table 1 that FDT-TM outperforms the BTG system significantly.

*FDT-DM* denotes the improved system that uses FDT-DM proposed in Section 3.3 instead of original distortion model. Comparing to baseline DM which has length limitation on training instances, training examples of FDT-DM are extracted from 1-

best FDTs without any restriction. This makes our new DM can cover both local and global reordering situations that might be met in decoding procedures. We can see from Table 1 that using FDT-DM, significant improvements can be achieved.

*FDT-SLM* denotes the improved system that uses FDT-SLM proposed in Section 3.4 as an additional feature, in which the maximum n-gram order is 4. However, from Table 1 we notice that with FDT-SLM integrated, only 0.2 BLEU improvements can be obtained. We analyze decoding-logs and find that the reordered source sequences of n-best translations are very similar, which, we think, can explain why improvements of using this model are so limited.

*FDT-RSM* denotes the improved system that uses FDT-RSM proposed in Section 3.5 as an additional feature. The maximum order of this model is 3. From Table 1 we can see that FDT-RSM outperforms BTG significantly, with up to 0.48 BLEU improvements. Comparing to FDT-SLM, FDT-RSM performs slightly better as well. We think it is due to the fact that bilingual phrases can provide more discriminative power than monolingual n-grams do.

Last, all these four FDT-based models (FDT-TM, FDT-DM, FDT-SLM and FDT-RSM) are put together to form an improved system that is denoted as *FDT-ALL*. It can provide an averaged 1.2 BLEU improvements on these three evaluation data sets.

BLEU	dev	test-1	test-2
BTG	20.60	20.27	13.15
FDT-TM	21.21	20.71(+0.44)	13.98(+0.83)
FDT-DM	21.13	20.79(+0.52)	14.25(+1.10)
FDT-SLM	20.84	20.50(+0.23)	13.36(+0.21)
FDT-RSM	21.07	20.75(+0.48)	13.59(+0.44)
FDT-ALL	<b>21.83</b>	<b>21.34(+1.07)</b>	<b>14.46(+1.31)</b>
PRO	21.89	21.81	14.69

Table 1: FDT-based model training on E-J task.

Pre-reordering (*PRO*) is often used on language pairs, e.g. English and Japanese, with very different word orders. So we compare our method with PRO as well. We re-implement the PRO method proposed by Genzel (2010) and show its results in Table 1. On dev and test-2, FDT-ALL performs comparable to PRO, with no syntactic information needed at all.

## 5.4 Translation Quality on C-E Task

We then evaluate the effectiveness of our FDT-based model training approach on C-E translation task, and present evaluation results in Table 2, from which we can see significant improvements as well.

BLEU	MT03	MT05	MT08
BTG	38.73	38.01	23.78
FDT-TM	39.14	38.31(+0.30)	24.30(+0.52)
FDT-DM	39.27	38.56(+0.55)	24.50(+0.72)
FDT-SLM	38.97	38.22(+0.21)	24.04(+0.26)
FDT-RSM	39.06	38.33(+0.32)	24.13(+0.35)
<b>FDT-ALL</b>	<b>39.59</b>	<b>38.72(+0.71)</b>	<b>24.67(+0.89)</b>

Table 2: FDT-based model training on C-E task

Comparing to numbers in Table 1, the gains coming from the first two FDT-based models become small on C-E task. This might be due to the fact that the word alignment quality in C-E task is more reliable than that in E-J task for TM and DM training.

## 5.5 Effect on Alignment Quality

We compare the qualities of alignments predicted by GIZA++ and alignments induced from 1-best FDTs.

For E-J task, 575 English-Japanese sentence pairs are manually annotated with word alignments. 382 sentence pairs are used as the dev set, and the other 193 sentence pairs are used as the test set. For C-E task, 491 Chinese-English sentence pairs are manually annotated with word alignments. 250 sentence pairs are used as the dev set, and the other 241 sentence pairs are used as the test set. Both Japanese and Chinese sentences are adapted to our own word segmentation standards respectively. Table 3 shows the comparison results. Comparing to C-E language pair (S-V-O), E-J language pair (S-O-V) has much lower F-scores, due to its very different word order.

F-score	from GIZA++	from 1-best FDTs
dev <sub>EJ</sub>	54.75%	57.93%(+3.18%)
test <sub>EJ</sub>	55.32%	58.47%(+3.15%)
dev <sub>CE</sub>	81.32%	83.37%(+2.05%)
test <sub>CE</sub>	80.61%	82.51%(+1.90%)

Table 3: Comparison of alignment qualities predicted by GIZA++ and induced from 1-best FDTs.

From Table 3 we can see that the F-score improves on all language pairs when using alignments induced from 1-best FDTs, rather than GIZA++.

## 5.6 Effect on Classification Accuracy

In the BTG system, the MaxEnt model is used as a binary classifier to predict reordering operations of neighbor translation blocks. As the baseline DM and our FDT-DM have different mechanisms on training instance extraction procedures, we compare the classification accuracies of these two DMs in Table 4 to show the effect of different training instances. The MaxEnt toolkit (Zhang, 2004) is used to optimize feature weights using the l-BFGS method (Byrd et al., 1995). We set the iteration number to 200 and Gaussian prior to 1 for avoiding overfitting. Table 4 shows that when using training instances extracted from FDTs, classification accuracy of reorderings improves on both E-J and C-E tasks. This is because FDTs can provide more deterministic and structured knowledge for training instance extraction, which can cover both local and global reordering cases.

	baseline DM	FDT-based DM
E-J	93.67%	95.60%(+1.93%)
C-E	95.85%	97.52%(+1.67%)

Table 4: Comparison of classification accuracies of DMs based on instances extracted by different mechanisms.

## 6 Conclusions

In this paper, we have presented an FDT-based model training approach to SMT. As the first step in this research direction, we have verified our method on a phrase-based SMT system, and proposed four FDT-based component models. Experiments on both E-J and C-E tasks have demonstrated the effectiveness of our approach. Summing up, comparing to plain word alignments, FDTs provide richer structured knowledge for more accurate SMT model training. Several potential research topics can be explored in future. For example, FDTs can be used in a pre-reordering framework. This is feasible in the sense that FDTs can provide both tree-like structures and reordering information. We also plan to adapt our FDT-based model training approach to SCFG-based and traditional left-to-right phrase-based systems.



## References

- Peter Brown, Stephen Pietra, Vincent Pietra, and Robert Mercer. 1993. *The Mathematics of Statistical Machine Translation: Parameter Estimation*, Computational Linguistics.
- Rafael Banchs, Josep Crego, Adrià Gispert, Patrik Lambert, and Jos Mario. 2005. *Statistical Machine Translation of Euparl Data by using Bilingual N-grams*, In Proceedings of the ACL Workshop on Building and Using Parallel Texts.
- Alexandra Birch, Phil Blunsom, and Miles Osborne. 2010. *Metrics for MT evaluation: Evaluating reordering*, Machine Translation.
- Alexandra Birch and Miles Osborne. 2011. *Reordering metrics for MT*, In Proceedings of the Association for Computational Linguistics.
- Richard Byrd, Peihuang Lu, Jorge Nocedal, and Ciyu Zhou. 1995. *A limited memory algorithm for bound constrained optimization*, SIAM Journal of Science and Statistical Computing.
- Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. *Syntax-based Language Models for Statistical Machine Translation*, MT Summit.
- Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. *Clause Restructuring for Statistical Machine Translation*, In Proceedings of the Association for Computational Linguistics.
- John DeNero and Jakob Uszkoreit. 2011. *Inducing Sentence Structure from Parallel Corpora for Reordering*, In Proceedings of the Conference on Empirical Methods in Natural Language Processing.
- Minwei Feng, Arne Mauser, and Hermann Ney. 2010. *A Source-side Decoding Sequence Model for Statistical Machine Translation*, In Proceedings of the Conference of the Association for Machine Translation.
- Alexander Fraser and Daniel Marcu. 2006. *Semi-Supervised Training for Statistical Word Alignment*, In Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics.
- Dmitriy Genzel. 2010. *Automatically learning source-side reordering rules for large scale machine translation*, In Proceedings of the Conference on Computational Linguistics.
- Liang Huang and David Chiang. 2005. *Better k-best Parsing*, In Proceedings of International Conference on Parsing Technologies.
- Young-Suk Lee, Bing Zhao, and Xiaoqiang Luo. 2010. *Constituent Reordering and Syntax Models for English-to-Japanese Statistical Machine Translation*, In Proceedings of the Conference on Computational Linguistics.
- Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. 2009. *Weighted Alignment Matrices for Statistical Machine Translation*, In Proceedings of the Conference on Empirical Methods in Natural Language Processing.
- Yang Liu, Qun Liu, and Shouxun Lin. 2010. *Discriminative Word Alignment by Linear Modeling*, Computational Linguistics.
- Robert Moore. 2005. *A Discriminative Framework for Bilingual Word Alignment*, In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing.
- Robert Moore, Wen-tau Yih, and Andreas Bode. 2006. *Improved Discriminative Bilingual Word Alignment*, In Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics.
- Haitao Mi, Liang Huang, and Qun Liu. 2008. *Forest-based Translation*, In Proceedings of the Association for Computational Linguistics.
- Galley Michel and Christopher D. Manning. 2008. *A Simple and Effective Hierarchical Phrase Reordering Model*, In Proceedings of the Conference on Empirical Methods in Natural Language Processing.
- Franz Och. 2003. *Minimum Error Rate Training in Statistical Machine Translation*, In Proceedings of the Association for Computational Linguistics.
- Franz Och and Hermann Ney. 2004. *The Alignment Template Approach to Statistical Machine Translation*, Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation*, In Proceedings of the Association for Computational Linguistics.
- Chris Quirk and Arul Menezes. 2006. *Do we need phrases? Challenging the conventional wisdom in Statistical Machine Translation*, In Proceedings of the North American Chapter of the Association for Computational Linguistics.
- Lane Schwartz. 2008. *Multi-Source Translation Methods*, In Proceedings of the Conference of the Association for Machine Translation.
- Wade Shen, Brian Delaney, Tim Anderson, and Ray Slyph. 2008. *The MIT-LL/AFRL IWSLT-2008 MT System*, International Workshop on Spoken Language Translation.
- David Talbot, Hideto Kazawa, Hiroshi Ichikawa, Jason Katz-Brown, Masakazu Seno, and Franz Och. 2011. *A lightweight evaluation framework for machine translation reordering*, In Proceedings of the Sixth Workshop on Statistical Machine Translation.
- Ashish Vaswani, Haitao Mi, Liang Huang, and David Chiang. 2011. *Rule Markov Models for Fast Tree-to-String Translation*, In Proceedings of the Association for Computational Linguistics.

- Ashish Venugopal, Andreas Zollmann, Noah Smith, and Stephan Vogel. 2008. *Wider Pipelines: N-best Alignments and Parses in MT Training*, In Proceedings of the Conference of the Association for Machine Translation.
- Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. *Training Phrase Translation Models with Leaving-One-Out*, In Proceedings of the Association for Computational Linguistics.
- Dekai Wu. 1997. *Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora*, Computational Linguistics.
- Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. *Maximum entropy based phrase reordering model for statistical machine translation*, In Proceedings of the Association for Computational Linguistics.
- Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. *Using a Dependency Parser to Improve SMT for Subject-Object-Verb Languages*, In Proceedings of the North American Chapter of the Association for Computational Linguistics.
- Richard Zens and Hermann Ney. 2006. *Discriminative Reordering Models for Statistical Machine Translation*, In Proceedings of the Workshop on Statistical Machine Translation.
- Le Zhang. 2004. *Maximum Entropy Modeling Toolkit for Python and C++*.