# Context-free reordering, finite-state translation

**Chris Dyer and Philip Resnik**
UMIACS Laboratory for Computational Linguistics and Information Processing
Department of Linguistics
University of Maryland, College Park, MD 20742, USA
redpony, resnik AT umd.edu

## Abstract

We describe a class of translation model in which a set of input variants encoded as a context-free forest is translated using a finite-state translation model. The forest structure of the input is well-suited to representing word order alternatives, making it straightforward to model translation as a two step process: (1) tree-based source reordering and (2) phrase transduction. By treating the reordering process as a latent variable in a probabilistic translation model, we can learn a long-range source reordering model without example reordered sentences, which are problematic to construct. The resulting model has state-of-the-art translation performance, uses linguistically motivated features to effectively model long range reordering, and is significantly smaller than a comparable hierarchical phrase-based translation model.

## 1 Introduction

Translation models based on synchronous context-free grammars (SCFGs) have become widespread in recent years (Wu, 1997; Chiang, 2007). Compared to phrase-based models, which can be represented as finite-state transducers (FSTs, Kumar et al. (2006)), one important benefit that SCFG models have is the ability to process long range reordering patterns in space and time that is polynomial in the length of the displacement, whereas an FST must generally explore a number of states that is exponential in this length.[1] As one would expect, for language pairs with substantial structural differences (and thus requiring long-range reordering during translation), SCFG models have come to outperform the best FST models (Zollmann et al., 2008).

In this paper, we explore a new way to take advantage of the computational benefits of CFGs during translation. Rather than using a single SCFG to both reorder and translate a source sentence into the target language, we break the translation process into a two step pipeline where (1) the source language is reordered into a target-like order, with alternatives encoded in a context-free forest, and (2) the reordered source is transduced into the target language using an FST that represents phrasal correspondences.

While multi-step decompositions of the translation problem have been proposed before (Kumar et al., 2006), they are less practical with the rise of SCFG models, since the context-free languages are not closed under intersection (Hopcroft and Ullman, 1979). However, the CFLs are closed under intersection with regular languages. By restricting ourselves to a *finite-state* phrase transducer and representing reorderings of the source in a *context-free* forest, exact inference over the composition of the two models is possible.

The paper proceeds as follows. We first explore reordering forests and describe how to translate them with an FST (§2). Since we would like our reordering model to discriminate between good reorderings of the source and bad ones, we show how to train our reordering component as a latent variable in an end-to-end translation model (§3). We then presents experimental results on language pairs requiring small amounts and large amounts of reordering (§4). We conclude with a discussion of related

[1] Our interest here is the reordering made possible by varying the arrangement of the translation units, not the local word order differences captured inside memorized phrase pairs.



*Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 858–866,
Los Angeles, California, June 2010. 

work (§6) and possible extensions (§7).

## 2 Reordering forests and translation

In this section, we describe *source reordering forests*, a context-free representation of source language word order alternatives.[2] The basic idea is that for the source sentence, **f**, that is to be translated, we want to create a (monolingual) context-free grammar $\mathcal{F}$ that generates strings ($\mathbf{f}'$) of words in the source language that are permutations of the original sentence. Specifically, this forest should contain derivations that put the source words into an order that approximates how they will be ordered in the grammar of the target language.

For a concrete example, let us consider the task of English-Japanese translation.[3] Our input sentence is *John ate an apple*. Japanese is a *head-final language*, where the heads of phrases (such as the verb in a verb phrase) typically come last, and English is a *head-initial language*, where heads come first. As a result, the usual order for a declarative sentence in English is SVO (subject-verb-object), but in Japanese, it is SOV, and the desired translation is *John-ga ringo-o* [an apple] *tabeta* [ate]. In summary, when translating from English into Japanese, it is usually necessary to move verbs from their position between the subject and object to the end of the sentence.

This reordering can happen in two ways, which we depict in Figure 1. In the derivation on the left, a memorized phrase pair captures the movement of the verb (Koehn et al., 2003). In the other derivation, the source is first reordered into target word order and then translated, using smaller translation units. In addition, we have assumed that the phrase translations were learned from a parallel corpus that is in the original ordering, so the reordering forest $\mathcal{F}$ should include derivations of phrase-size units in the source order as well as the target order.

[2]Note that forests are isomorphic to context-free grammars. For example, what is referred to as the 'parse forest', and understood to encode all derivations of a sentence **s** under some grammar, can also be understood as being a context-free grammar itself that exactly generates **s**. We therefore refer to a forest as a grammar sometimes, or vice versa, depending on which characterization is clearer in context.

[3]We use English as the source language since we expect the parse structure of English sentences will be more familiar to many readers.

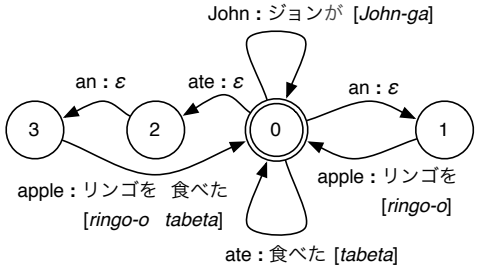


Figure 2: A fragment of a phrase-based English-Japanese translation model, represented as an FST. Japanese romanization is given in brackets.

A minimal reordering forest that supports the derivations depicted needs to include both an SOV and SVO version of the source. This could be accomplished trivially with the following grammar:

$$\begin{aligned} \text{S} &\rightarrow \textit{John ate an apple} \\ \text{S} &\rightarrow \textit{John an apple ate} \end{aligned}$$

However, this grammar misses the opportunity to take advantage of the regularities in the permuted structure. A better alternative might be:

$$\begin{aligned} \text{S} &\rightarrow \textit{John}\ \text{VP} \\ \text{VP} &\rightarrow \textit{ate}\ \text{NP} \\ \text{VP} &\rightarrow \text{NP}\ \textit{ate} \\ \text{NP} &\rightarrow \textit{an apple} \end{aligned}$$

In this grammar, the phrases *John* and *an apple* are fixed and only the VP contains ordering ambiguity.

### 2.1 Reordering forests based on source parses

Many kinds of reordering forests are possible; in general, the best one for a particular language pair will be one that is easiest to create given the resources available in the source language. It will also be the one that most compactly expresses the source reorderings that are most likely to be useful for translation. In this paper, we consider a particular kind of reordering forest that is inspired by the reordering model of Yamada and Knight (2001).[4] These are generated by taking a source language parse tree and 'expanding' each node so that it

[4]One important difference is that our translation model is not restricted by the structure of the source parse tree; i.e., phrases used in transduction need not correspond to constituents in the source reordering forest. However, if a phrase does cross a constituent boundary between constituents $A$ and $B$, then translations that use that phrase will have $A$ and $B$ adjacent.

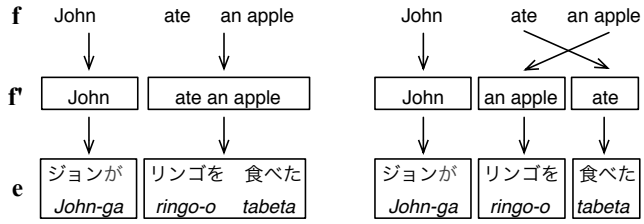


Figure 1: Two possible derivations of a Japanese translation of an English source sentence.

rewrites with different permutations of its children.[5]

For an illustration using our example sentence, refer to Figure 3 for the forest representation and Figure 4 for its isomorphic CFG representation. It is easy to see that this forest generates the two 'good' order variants from Figure 1; however, the forest includes many other derivations that will probably not lead to good translations. For this reason, it is helpful to associate the edges in the forest (that is, the rules in the CFG) with weights reflecting how likely that rule is to lead to a good translation. We discuss how these weights can be learned automatically in §3.

### 2.2 Translating reordering forests with FSTs

Having described how to construct a context-free reordering forest for the source sentence, we now turn to the problem of how to translate the source forest into the target language using a phrase-based translation model encoded as an FST, e.g. Figure 2. The process is quite similar to the one used when translating a source sentence with an SCFG, but with a twist: rather than representing the translation model as a grammar and parsing the source sentence, we represent the *source sentence* as a grammar (i.e. its reordering forest), and we use it to 'parse' the *translation model* (i.e. the FST representation of the phrase-based model). The end result (either way!) is a translation forest containing all possible target-language translations of the source.

Parsing can be understood as a means of computing the intersection of an FSA and a CFG (Grune and Jacobs, 2008). Since we are dealing with FSTs that define binary relations over strings, not FSAs defining strings, this operation is more properly *composition*. However, since CFG/FSA intersection is less cumbersome to describe, we present the algorithm in terms of intersection.

To compute the composition of a reordering forest, $\mathcal{G}$, with an FSA, $F$, we will make use of a variant of Earley's algorithm (Earley, 1970). Let weighted finite-state automaton $F = \langle \Sigma, Q, q_0, q_{\mathit{final}}, \delta, w \rangle$. $\Sigma$ is a finite alphabet; $Q$ is a set of states; $q_0$ and $q_{\mathit{final}} \in Q$ are start and accept states, respectively,[6] $\delta$ is the transition function $Q \times \Sigma \rightarrow 2^Q$, and $w$ is the transition cost function $Q \times Q \rightarrow \mathbb{R}$. We use variables that refer to states in the FSA with the letters $q$, $r$, and $s$. We use $x$ to represent a variable that is an element of $\Sigma$. Variables $u$ and $v$ represent costs. $X$ and $Y$ are non-terminals. Lowercase Greek letters are strings of terminals and non-terminals. The function $\delta(q, x)$ returns the state(s) that are reachable from state $q$ by taking a transition labeled with $x$ in the FSA.

Figure 5 provides the inference rules for a top-down intersection algorithm in the form of a weighted logic program; the three inference rules correspond to Earley's SCAN, PREDICT, and COMPLETE, respectively.

## 3 Reordering and translation model

As pointed out in §2.1, our reordering forests may contain many paths, some of which when translated will lead to good translations and others that will be bad. We would like a model to distinguish the two.

If we had a parallel corpus of source language sentences paired with 'reference reorderings', such a model could be learned directly as a supervised learning task. However, creating the optimal target-language reordering $\mathbf{f}'$ for some $\mathbf{f}$ is a nontrivial task.[7] Instead of trying to solve this problem, we opt to treat the reordered from of the source, $\mathbf{f}'$, as a

[5] For computational tractability, we only consider all permutations only when the number of children is less than 5, otherwise we exclude permutations where a child moves more than 4 positions away from where it starts.

[6] Other FSA definitions permit sets of start and final states. We use the more restricted definition for simplicity and because in our FSTs $q_0 = q_{\mathit{final}}$.

[7] For a discussion of methods for generating reference re-

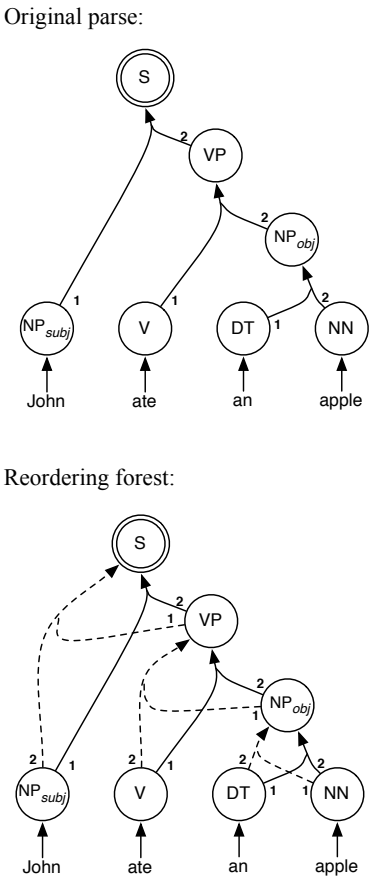


Figure 3: Example of a reordering forest. Linearization order of non-terminals is indicated by the index at the tail of each edge. The isomorphic CFG is shown in Figure 4; dashed edges correspond to reordering-specific rules.

| | |
|---|---|
| Original parse grammar: | $S \rightarrow NP_{subj}\ VP$ |
| $VP \rightarrow V\ NP_{obj}$ | $NP_{obj} \rightarrow DT\ NN$ |
| $NP_{subj} \rightarrow$ *John* | $V \rightarrow$ *ate* |
| $DT \rightarrow$ *an* | $NN \rightarrow$ *apple* |

Additional reordering grammar rules:

$$S \rightarrow VP\ NP_{subj}$$
$$VP \rightarrow NP_{obj}\ V$$
$$NP_{obj} \rightarrow NN\ DT$$

Figure 4: Context-free grammar representation of the forest in Figure 3. The reordering grammar contains the parse grammar, plus the reordering-specific rules.

Initialization:

$$\overline{[S' \rightarrow \bullet S, q_0, q_0] : \overline{1}}$$

Inference rules:

$$\frac{[X \rightarrow \alpha \bullet x\beta, q, r] : u}{[X \rightarrow \alpha x \bullet \beta, q, \delta(r, x)] : u \otimes w(\delta(r, x))}$$

$$\frac{[X \rightarrow \alpha \bullet Y\beta, q, r]}{[Y \rightarrow \bullet\gamma, r, r] : u} \quad Y \xrightarrow{u} \gamma \in \mathcal{G}$$

$$\frac{[X \rightarrow \alpha \bullet Y\beta, q, s] : u \quad [Y \rightarrow \gamma\bullet, s, r] : v}{[X \rightarrow \alpha Y \bullet \beta, q, r] : u \otimes v}$$

Goal state:

$$[S' \rightarrow S\bullet, q_0, q_{final}]$$

Figure 5: Weighted logic program for computing the intersection of a weighted FSA and a weighted CFG.

*latent variable* in a probabilistic translation model. By doing this, we only require a parallel corpus of translations to learn the reordering model. Not only does this make our lives easier, since 'reference reorderings' are not necessary, but it is also intuitively satisfying because from a task perspective, we are not concerned with values of $\mathbf{f}'$, but only with producing a good translation $\mathbf{e}$.

### 3.1 A probabilistic translation model with a latent reordering variable

The translation model we use is a two phase process. First, source sentence $\mathbf{f}$ is reordered into a target-like word order $\mathbf{f}'$ according to a reordering model $r(\mathbf{f}'|\mathbf{f})$. The reordered source is then transduced into the target language according to a translation model $t(\mathbf{e}|\mathbf{f}')$. We require that $r(\mathbf{f}'|\mathbf{f})$ can be represented by a recursion-free probabilistic context-free grammar, i.e. a forest as in §2.1, and that $t(\mathbf{e}|\mathbf{f}')$ is represented by a (cyclic) finite-state transducer, as in Figure 2.

Since the reordering forest may define multiple derivations $\mathbf{a}$ from $\mathbf{f}$ to a particular $\mathbf{f}'$, and the transducer may define multiple derivations $\mathbf{d}$ from $\mathbf{f}'$ to a particular translation $\mathbf{e}$, we marginalize over these nuisance variables as follows to define the probability of a translation given the source:

$$p(\mathbf{e}|\mathbf{f}) = \sum_{\mathbf{d}} \sum_{\mathbf{f}'} t(\mathbf{e}, \mathbf{d}|\mathbf{f}') \sum_{\mathbf{a}} r(\mathbf{f}', \mathbf{a}|\mathbf{f}) \quad (1)$$

Crucially, since we have restricted $r(\mathbf{f}'|\mathbf{f})$ to have the form of a weighted CFG and $t(\mathbf{e}|\mathbf{f}')$ to be an

orderings from word aligned parallel corpora, refer to Tromble and Eisner (2009).

FST, the quantity (1), which sums over all reorderings (and derivations), can be computed in polynomial time with dynamic programming composition, as described in §2.2.

### 3.2 Conditional training

While it is straightforward to use expectation maximization to optimize the joint likelihood of the parallel training data with a latent variable model, instead we use a log-linear parameterization and maximize conditional likelihood (Blunsom et al., 2008; Petrov and Klein, 2008). This enables us to employ a rich set of (possibly overlapping, non-independent) features to discriminate among translations. The probability of a derivation from source to reordered source to target is thus written in terms of model parameters $\Lambda = \{\lambda_i\}$ as:

$$p(\mathbf{e}, \mathbf{d}, \mathbf{f}', \mathbf{a}|\mathbf{f}; \Lambda) = \frac{\exp \sum_i \lambda_i \cdot H_i(\mathbf{e}, \mathbf{d}, \mathbf{f}', \mathbf{a}, \mathbf{f})}{Z(\mathbf{f}; \Lambda)}$$

$$\text{where } H_i(\mathbf{e}, \mathbf{d}, \mathbf{f}', \mathbf{a}, \mathbf{f}) = \sum_{r \in \mathbf{d}} h_i(\mathbf{f}', r) + \sum_{s \in \mathbf{a}} h_i(\mathbf{f}, s)$$

The derivation probability is globally normalized by the partition $Z(\mathbf{f}; \Lambda)$, which is just the sum of the numerator for all derivations of $\mathbf{f}$ (corresponding to any $\mathbf{e}$). The $H_i$ (written below without their arguments) are real-valued feature functions that may be overlapping and non-independent. For computational tractability, we assume that the feature functions $H_i$ decompose with the derivations of $\mathbf{f}'$ and $\mathbf{e}$ in terms of *local* feature functions $h_i$. We also define $Z(\mathbf{e}, \mathbf{f}; \lambda)$ to be the sum of the numerator over all derivations that yield the sentence pair $\langle \mathbf{e}, \mathbf{f} \rangle$. Rather than training purely to optimize conditional likelihood, we also make use of a spherical Gaussian prior on the value of $\Lambda$ with mean $\mathbf{0}$ and variance $\sigma^2$, which helps prevent overfitting of the model (Chen and Rosenfeld, 1998). Our objective is thus to select $\Lambda$ minimizing:

$$\begin{aligned}\mathcal{L} &= -\log \prod_{\langle \mathbf{e}, \mathbf{f} \rangle} p(\mathbf{e}|\mathbf{f}; \Lambda) - \frac{||\Lambda||^2}{2\sigma^2} \\ &= -\sum_{\langle \mathbf{e}, \mathbf{f} \rangle} [\log Z(\mathbf{e}, \mathbf{f}; \Lambda) - \log Z(\mathbf{f}; \Lambda)] - \frac{||\Lambda||^2}{2\sigma^2}\end{aligned}$$

The gradient of $\mathcal{L}$ with respect to the feature weights has a parallel form; it is the difference in feature expectations under the reference distribution and the translation distribution with a penalty term due to the prior:

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = \sum_{\langle \mathbf{e}, \mathbf{f} \rangle} \mathbb{E}_{p(\mathbf{d}, \mathbf{a}|\mathbf{e}, \mathbf{f}; \Lambda)}[h_i] - \mathbb{E}_{p(\mathbf{e}, \mathbf{d}, \mathbf{a}|\mathbf{f}; \Lambda)}[h_i] - \frac{\lambda_i}{\sigma^2}$$

The form of the objective and gradient are quite similar to the traditional fully observed training scenario for CRFs (Sha and Pereira, 2003). However, rather than matching the feature expectations in the model to an observable feature value, we have to sum over the latent structure that remains after observing our target $\mathbf{e}$, which makes the form of the first summand an expectation rather than just a feature function value.

#### 3.2.1 Computing the objective and gradient

The objective and gradient that were just introduced can be computed in two steps. Given a training pair $\langle \mathbf{e}, \mathbf{f} \rangle$, we generate the forest of reorderings $\mathcal{F}$ from $\mathbf{f}$ as described in §2.1. We then compose this grammar with $T$, the FST representing the translation model, which yields $\mathcal{F} \circ T$, a translation forest that contains all possible translations of $\mathbf{f}$ into the target language, as described in §2.2. Running the inside algorithm on the translation forest computes $Z(\mathbf{f}; \Lambda)$, the first term in the objective, and the inside-outside algorithm can be used to compute $\mathbb{E}_{p(\mathbf{e}, \mathbf{d}, \mathbf{a}|\mathbf{f})}[h_i]$. Next, to compute $Z(\mathbf{e}, \mathbf{f}; \Lambda)$ and the first expectation in the gradient, we need to find the subset of the translation forest $\mathcal{F} \circ T$ that exactly derives the reference translation $\mathbf{e}$. To do this, we again rely on the fact that $\mathcal{F} \circ T$ is a forest and therefore itself a context-free grammar. So, we use *this* grammar to parse the target reference string $\mathbf{e}$. The resulting forest, $\mathcal{F} \circ T \circ \mathbf{e}$, contains all and only derivations that yield the pair $\langle \mathbf{e}, \mathbf{f} \rangle$. Here, the inside algorithm computes $Z(\mathbf{e}, \mathbf{f}; \Lambda)$ and the inside-outside algorithm can be used to compute $\mathbb{E}_{p(\mathbf{e}, \mathbf{d}, \mathbf{a}|\mathbf{f})}[h_i]$.

Once we have an objective and gradient, we can apply any first-order numerical optimization technique.[8] Although the conditional likelihood surface of this model is non-convex (on account of the latent variables), we did not find a significant initialization effect. For the experiments below, we initialized $\Lambda = \mathbf{0}$ and set $\sigma^2 = 1$. Training generally converged in fewer than 1500 function evaluations.

[8] For our experiments we used L-BFGS (Liu and Nocedal, 1989).

## 4 Experimental setup

We now turn to an experimental validation of the models we have introduced. We define three conditions: a small data scenario consisting of a translation task based on the BTEC Chinese-English corpus (Takezawa et al., 2002), a large data Chinese-English condition designed to be more comparable to conditions in a NIST MT evaluation, and a large data Arabic-English task.

For each condition, phrase tables were extracted as described in Koehn et al. (2003) with a maximum phrase size of 5. The parallel training data was aligned using the Giza++ implementation of IBM Model 4 (Och and Ney, 2003). The Chinese text was segmented using a CRF-based word segmenter (Tseng et al., 2005). The Arabic text was segmented using the technique described in Lee et al. (2003). The Stanford parser was used to generate source parses for all conditions, and these were then used to generate the reordering forests as described in §2.1.

Table 1 summarizes statistics about the corpora used. The reachability statistic indicates what percentage of sentence pairs in the training data could be regenerated using our reordering/translation model.[9] To train the reordering model, we used all of the reachable sentence pairs from BTEC, 20% of the reachable set in the Chinese-English condition, and all reachable sentence pairs under 40 words (source) in length in the Arabic-English condition.

Error analysis indicates that a substantial portion of unreachable sentence pairs are due to alignment (word or sentence) or parse errors; however, in some cases the reordering forests did not contain an adequate source reordering to produce the necessary target. For example, in Arabic, which is a VSO language, the treebank annotation is to place the subject NP as the 'middle child' between the V and the object constituent. This can be reordered into an English SVO order using our child-permutation rules; however, if the source VP is modified by a modal particle, the parser makes the particle the parent of the VP, and it is no longer possible to move the subject to the first position in the sentence. Richer reordering rules are needed to address this problem. Other solutions to the reachability problem include targeting reachable oracles instead of the reference translation (Li and Khudanpur, 2009) or making use of alternative training criteria, such as minimum risk training (Li and Eisner, 2009).

[9] Only sentences that can be generated by the model can be used in training.

### 4.1 Features

We briefly describe the feature functions we used in our model. These include the typical dense features used in translation: relative phrase translation frequencies $p(\overline{e}|\overline{f})$ and $p(\overline{f}|\overline{e})$, 'lexically smoothed' translation probabilities $p_{lex}(\overline{e}|\overline{f})$ and $p_{lex}(\overline{f}|\overline{e})$, and a phrase count feature. For the reordering model, we used a binary feature for each kind of rule used, for example $\phi_{\text{VP}\rightarrow\text{V NP}}(\mathbf{a})$ would fire once for each time the rule VP $\rightarrow$ V NP was used in a derivation, **a**. For the Arabic-English condition, we observed that the parse trees tended to be quite flat, with many repeated non-terminal types in one rule, so we augmented the non-terminal types with an index indicating where they were located in the original parse tree. This resulted in a total of 6.7k features for IWSLT, 18k features for the large Chinese-English condition, and 516k features for Arabic-English.[10] A target language model was not used during the training of the source reordering model, but it was used during the translation experiments (see below).

### 4.2 Qualitative assessment of reordering model

Before looking at the translation results, we examine what the model learns during training. Figure 6 lists the 10 most highly weighted reordering features learned by the BTEC model (above) and shows an example reordering using this model (below), with the most English-like reordering indicated with a star.[11] Keep in mind, we expect these features to reflect what the best *English-like* order of the input should be. All are almost surprisingly intuitive, but this is not terribly surprising since Chinese and English have very similar large-scale structures (both are head initial, both have adjectives and quantifiers that precede nouns). However, we see two entries in the list (starred) that correspond to an En-

[10] The large number of features in the Arabic system was due to the relative flatness of the Arabic parse trees.

[11] The italicized symbols in the English gloss are functional elements with no precise translation. Q is an interrogative particle, and DE marks a variety of attributive roles and is used here as the head of a relative clause.

Table 1: Corpus statistics

| Condition | Sentences | Source words | Target words | Reachability |
|---|---|---|---|---|
| BTEC | 44k | 0.33M | 0.36M | 81% |
| Chinese-English | 400k | 9.4M | 10.9M | 25% |
| Arabic-English | 120k | 3.3M | 3.6M | 66% |

glish word order that is ungrammatical in Chinese: PP modifiers in Chinese typically precede the VPs they modify, and CPs (relative clauses) also typically precede the nouns they modify. In English, the reverse is true, and we see that the model has indeed learned to prefer this ordering. It was not necessary that this be the case: since our model makes use of phrases memorized from a non-reordered training set, it could hav relied on those for all its reordering. Yet these results provide evidence that it is learning large-scale reordering successfully.

| Feature | $\lambda$ | note |
|---|---|---|
| VP → VE NP | 0.995 | |
| VP → VV VP | 0.939 | modal + VP |
| VP → VV NP | 0.895 | |
| VP → VP PP* | 0.803 | PP modifier of VP |
| VP → VV NP IP | 0.763 | |
| PP → P NP | 0.753 | |
| IP → NP VP PU | 0.728 | PU = punctuation |
| VP → VC NP | 0.598 | |
| NP → DP NP | 0.538 | |
| NP → NP CP* | 0.537 | rel. clauses follow |

Input:

我 能 赶上 去 西尔顿 饭店 的 巴士 吗 ?

I CAN CATCH [NP[CP GO HILTON HOTEL ***DE***] BUS] ***Q*** ?

*(Can I catch a bus that goes to the Hilton Hotel ?)*

5-best reordering:

I CAN CATCH [NP BUS [CP GO HILTON HOTEL ***DE***]] ***Q*** ?

★ I CAN CATCH [NP BUS [CP ***DE*** GO HILTON HOTEL]] ***Q*** ?

I CAN CATCH [NP BUS [CP GO HOTEL HILTON ***DE***]] ***Q*** ?

I CATCH [NP BUS [CP GO HILTON HOTEL ***DE***]] CAN ***Q*** ?

I CAN CATCH [NP BUS [CP ***DE*** GO HOTEL HILTON]] ***Q*** ?

Figure 6: (Above) The 10 most highly-weighted features in a Chinese-English reordering model. (Below) Example reordering of a Chinese sentence (with English gloss, translation, and partial syntactic information).

## 5 Translation experiments

We now consider how to apply this model to a translation task. The training we described in §3.2 is suboptimal for state-of-the-art translation systems, since (1) it optimizes likelihood rather than an MT metric and (2) it does not include a language model. We describe how we addressed these problems here, and then present our results in the three conditions defined above.

### 5.1 Training for Viterbi decoding

A language model was incorporated using cube pruning (Huang and Chiang, 2007), using a 200-best limit at each node during LM integration. To improve the ability of the phrase model to match reordered phrases, we extracted the 1-best reordering of the training data under the learned reordering model and generated the phrase translation model so that it contained phrases from both the original order and the 1-best reordering.

To be competitive with other state-of-the-art systems, we would like to use Och's minimum error training algorithm for training; however, we cannot tune the model as described with it, since it has far too many features. To address this, we converted the coefficients on the reordering features into a single reordering feature which then had a coefficient assigned to it. This technique is similar to what is done with logarithmic opinion pools, only the learned model is not a probability distribution (Smith et al., 2005). Once we collapsed the reordering weights into a single feature, we used the techniques described by Kumar et al. (2009) to optimize the feature weights to maximize corpus BLEU on a held-out development set.

### 5.2 Translation results

Scores on a held-out test set are reported in Table 2 using case-insensitive BLEU with 4 reference translations (16 for BTEC) using the original definition of the brevity penalty. We report the results of our

model along with three baseline conditions, one with no-reordering at all (mono), the performance of a phrase-based translation model with distance-based distortion, the performance of our implementation of a hierarchical phrase-based translation model (Chiang, 2007), and then our model.

Table 2: Translation results (BLEU)

| Condition | Mono | PB | Hiero | Forest |
|---|---|---|---|---|
| BTEC | 47.4 | 51.8 | 52.4 | **54.1** |
| Chinese-Eng. | 29.0 | 30.9 | 32.1 | **32.4** |
| Arabic-Eng. | 41.2 | 45.8 | **46.6** | 44.9 |

## 6 Related work

A variety of translation processes can be formalized as the composition of a finite-state representation of input (typically just a sentence, but often a more complex structure, like a word lattice) with an SCFG (Wu, 1997; Chiang, 2007; Zollmann and Venugopal, 2006). Like these, our work uses parsing algorithms to perform the composition operation. But this is the first time that the *input* to a finite-state transducer has a context-free structure.[12] Although not described in terms of operations over formal languages, the model of Yamada and Knight (2001) can be understood as an instance of our class of models with a specific input forest and phrases restricted to match syntactic constituents.

In terms of formal similarity, Mi et al. (2008) use forests as input to a tree-to-string transducer process, but the forests are used to recover from 1-best parsing errors (as such, all derivations yield the same source string). Iglesias et al. (2009) use a SCFG-based translation model, but implement it using FSTs, although they use non-regular extensions that make FSTs equivalent to recursive transition networks. Galley and Manning (2008) use a context-free reordering model to score a phrase-based (exponential) search space.

Syntax-based preprocessing approaches that have relied on hand-written rules to restructure source trees for particular translation tasks have been quite widely used (Collins et al., 2005; Wang et al., 2007; Xu et al., 2009; Chang et al., 2009). Discriminatively trained reordering models have been extensively explored. A widely used approach has been to use a classifier to predict the orientation of phrases during decoding (Zens and Ney, 2006; Chang et al., 2009). These classifiers must be trained independently from the translation model using training examples extracted from the training data. A more ambitious approach is described by Tromble and Eisner (2009), who build a global reordering model that is learned automatically from reordered training data.

The latent variable discriminative training approach we describe is similar to the one originally proposed by Blunsom et al. (2008).

[12] Satta (submitted) discusses the theoretical possibility of this sort of model but provides no experimental results.

## 7 Discussion and conclusion

We have described a new model of translation that takes advantage of the strengths of context-free modeling, but splits reordering and phrase transduction into two separate models. This lets the context-free part handle what it does well, mid-to-long range reordering, and lets the finite-state part handle local phrasal correspondences. We have further shown that the reordering component can be trained effectively as a latent variable in a discriminative translation model using only conventional parallel training data.

This model holds considerable promise for future improvement. Not only does it already achieve quite reasonable performance (performing particularly well in Chinese-English, where mid-range reordering is often required), but we have only begun to scratch the surface in terms of the kinds of features that can be included to predict reordering, as well as the kinds of reordering forests used. Furthermore, by reintroducing the concept of a cascade of transducers into the context-free model space, it should be possible to develop new and more effective rescoring mechanisms. Finally, unlike SCFG and phrase-based models, our model does not impose any distortion limits.

## Acknowledgements

The authors gratefully acknowledge partial support from the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-2-001. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors. Thanks to Hendra Setiawan, Vlad Eidelman, Zhifei Li, Chris Callison-Burch, Brian Dillon and the anonymous reviewers for insightful comments.

## References


P. Blunsom, T. Cohn, and M. Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-HLT*.

P.-C. Chang, D. Jurafsky, and C. D. Manning. 2009. Disambiguating "DE" for Chinese-English machine translation,. In *Proc. WMT*.

S. F. Chen and R. Rosenfeld. 1998. A Gaussian prior for smoothing maximum entropy models. Technical Report TR-10-98, Computer Science Group, Harvard University.

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

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL 2005*.

J. Earley. 1970. An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery*, 13(2):94–102.

M. Galley and C. D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proc. EMNLP*.

D. Grune and C. J. H. Jacobs. 2008. Parsing as intersection. In D. Gries and F. B. Schneider, editors, *Parsing Techniques*, pages 425–442. Springer, New York.

J. E. Hopcroft and J. D. Ullman. 1979. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley.

L. Huang and D. Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *ACL*.

G. Iglesias, A. de Gispert, E. R. Banga, and W. Byrne. 2009. Hierarchical phrase-based translation with weighted finite state transducers. In *Proc. NAACL*.

P. Koehn, F. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL*, pages 48–54.

S. Kumar, Y. Deng, and W. Byrne. 2006. A weighted finite state transducer translation template model for statistical machine translation. *Journal of Natural Language Engineering*, 12(1):35–75.

S. Kumar, W. Macherey, C. Dyer, and F. Och. 2009. Efficient minimum error rate training and minimum Bayes-risk decoding for translation hypergraphs and lattices. In *Proc. ACL*.

Y.-S. Lee, K. Papineni, S. Roukos, O. Emam, and H. Hassan. 2003. Language model based Arabic word segmentation. In *Proc. ACL*.

Z. Li and J. Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proc. EMNLP*.

Z. Li and S. Khudanpur. 2009. Efficient extraction of oracle-best translations from hypergraphs. In *Proc. NAACL*.

D. C. Liu and J. Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

H. Mi, L. Huang, and Q. Liu. 2008. Forest-based translation. In *Proceedings of ACL-08: HLT*, pages 192–199, Columbus, Ohio, June. Association for Computational Linguistics.

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

S. Petrov and D. Klein. 2008. Discriminative log-linear grammars with latent variables. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 1153–1160.

G. Satta. submitted. Translation algorithms by means of language intersection.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL*, pages 213–220.

A. Smith, T. Cohn, and M. Osborne. 2005. Logarithmic opinion pools for conditional random fields. In *Proc. ACL*.

T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of LREC 2002*, pages 147–152, Las Palmas, Spain.

R. Tromble and J. Eisner. 2009. Learning linear order problems for better translation. In *Proceedings of EMNLP 2009*.

H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning. 2005. A conditional random field word segmenter. In *Fourth SIGHAN Workshop on Chinese Language Processing*.

C. Wang, M. Collins, and P. Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proc. EMNLP*.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404.

P. Xu, J. Kang, M. Ringgaard, and F. Och. 2009. Using a dependency parser to improve SMT for subject-object-verb languages. In *Proc. NAACL*, pages 245–253.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *Proc. ACL*.

R. Zens and H. Ney. 2006. Discriminative reordering models for statistical machine translation. In *Proc. of the Workshop on SMT*.

A. Zollmann and A. Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proc. of the Workshop on SMT*.

A. Zollmann, A. Venugopal, F. Och, and J. Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proc. Coling*.