Prior Derivation Models For Formally Syntax-based Translation Using Linguistically Syntactic Parsing and Tree Kernels

Bowen Zhou
IBM T. J. Watson Research Center
Yorktown Heights, NY
zhou@us.ibm.com

Bing Xiang
IBM T. J. Watson Research Center
Yorktown Heights, NY
bxiang@us.ibm.com

Xiaodan Zhu
Dept. of Computer Science
University of Toronto
xzhu@cs.toronto.edu

Yuqing Gao
IBM T. J. Watson Research Center
Yorktown Heights, NY
yuqing@us.ibm.com

Abstract

This paper presents an improved formally syntax-based SMT model, which is enriched by linguistically syntactic knowledge obtained from statistical constituent parsers. We propose a linguistically-motivated prior derivation model to score hypothesis derivations on top of the baseline model during the translation decoding. Moreover, we devise a fast training algorithm to achieve such improved models based on tree kernel methods. Experiments on an English-to-Chinese task demonstrate that our proposed models outperformed the baseline formally syntax-based models, while both of them achieved significant improvements over a state-of-the-art phrase-based SMT system.

1 Introduction

In recent years, syntax-based translation models (Chiang, 2007; Galley et al., 2004; Liu et al., 2006) have shown promising progress in improving translation quality. There are two major elements accounting for such an improvement: namely the incorporation of phrasal translation structures adopted from widely applied phrase-based models (Och and Ney, 2004) to handle local fluency, and the engagement of synchronous context-free grammars (SCFG), which enhances the generative capacity of the underlying model that is limited by finite-state machinery.

Approaches to syntax-based translation models using SCFG can be further categorized into two classes, based on their dependency on annotated corpus. Following Chiang (Chiang, 2007), we note the following distinction between these two classes:

- *Linguistically* syntax-based: models that utilize structures defined over linguistic theory and annotations (e.g., Penn Treebank), and SCFG rules are derived from parallel corpus that is guided by explicitly parsing on at least one side of the parallel corpus. Examples among others are (Yamada and Knight, 2001) and (Galley et al., 2004).

- *Formally* syntax-based: models are based on hierarchical structures of natural language but synchronous grammars are automatically extracted from parallel corpus without any usage of linguistic knowledge or annotations. Examples include Wu's (Wu, 1997) ITG and Chiang's hierarchical models (Chiang, 2007).

While these two often resemble in appearance, from practical viewpoints, there are some distinctions in training and decoding procedures differentiating formally syntax-based models from linguistically syntax-based models. First, the former has no dependency on available linguistic theory and annotations for targeting language pairs, and thus the training and rule extraction are more efficient. Secondly, the decoding complexity of the former is lower [1], especially when integrating a n-gram based

[1] The complexity is dominated by synchronous parsing and boundary words keeping. Thus binary SCFG employed in formally syntax-based systems help to maintain efficient CKY decoding. Recent work by (Zhang et al., 2006) shows a practically efficient approach that binarizes linguistically SCFG rules when possible.

Proceedings of the Second ACL Workshop on Syntax and Structure in Statistical Translation (SSST-2), pages 19–27,
ACL-08: HLT, Columbus, Ohio, USA, June 2008.

language model, which is a key element to ensure translation output quality.

On the other hand, available linguistic theory and annotations could provide invaluable benefits in grammar induction and scoring, as shown by recent progress on such models (Galley et al., 2006). In contrast, formally syntax-based grammars often lack explicit linguistic constraints.

In this paper, we propose a scheme to enrich formally syntax-based models with linguistically syntactic knowledge. In other words, we maintain our grammar to be based on formal syntax on surface, but incorporate linguistic knowledge into our models to leverage syntax theory and annotations.

Our goal is two-fold. First, how to score SCFG rules whose general abstraction forms are unseen in the training data is an important question to answer. In hierarchical models, Chiang (Chiang, 2007) utilizes heuristics where certain assumptions are made on rule distributions to obtain relative frequency counts. We intend to explore if additional linguistically parsing information would be beneficial to improve the scoring of formally syntactic SCFG grammars. Secondly, we note that SCFG-based models often come with an excessive memory consumption as its rule size is an order of magnitude larger compared to phrase-based models, which challenges its practical deployment for online real-time translation tasks. Furthermore, formal syntax rules are often redundant as they are automatically extracted without linguistic supervision. Therefore, we are motivated to study approaches to further score and rank formal syntax rules based on syntax-inspired methods, and eventually to prune unnecessary rules without loss of performance in general.

In our study, we propose a linguistically-motivated method to train prior derivation models for formally syntax-based translation. In this framework, prior derivation models can be viewed as a smoothing of rule translation models, addressing the weakness of the baseline model estimation that relies on relative counts obtained from heuristics. First, we apply automatic parsers to obtain syntax annotations on the English side of the parallel corpus. Next, we extract tree fragments associated with phrase pairs, and measure similarity between such tree fragments using kernel methods (Collins and Duffy, 2002; Moschitti, 2006). Finally, we score and rank rules based on their minimal cluster similarity of their nonterminals, which is used to compute the prior distribution of hypothesis derivations during decoding for improved translation.

The remainder of the paper is organized as follows. We start with a brief review of some related work in Sec. 2. In Sec. 3, we describe our formally syntax-based models and decoder implementation, that is established as our baseline system. Sec. 4 presents the approach to score formal SCFG rules using kernel methods. Experimental results are provided in Sec. 5. Finally, Sec. 6 summarized our contributions with discussions and future work.

2 Related Work

Syntax-based translation models engaged with SCFG have been actively investigated in the literature (Wu, 1997; Yamada and Knight, 2001; Gildea, 2003; Galley et al., 2004; Satta and Peserico, 2005). Recent work by (Chiang, 2007; Galley et al., 2006) shows promising improvements compared to phrase-based models for large-scale tasks. However, few previous work directly applied linguistically syntactic information into a formally syntax-based models, which is explored in this paper.

Kernel methods leverage the fact that the only operation in a procedure is the evaluation of inner dot products between pairs of observations, where the inner product is thus replaced with a Mercer kernel that provides an efficient way to carry out computation when original feature dimension is large or even infinite. Collins and Duffy (Collins and Duffy, 2002) suggested to employ convolution kernels to measure similarity between two trees in terms of their substructures, and more recently, Moschitti (Moschitti, 2006) described in details a fast implementation of tree kernels. To our knowledge, this paper is one of the few efforts of applying kernel methods for improved translation.

3 Formally Syntax-based Models

An SCFG is a synchronous rewriting system generating source and target side string pairs simultaneously based on context-free grammar. Each synchronous production (i.e., rule) rewrites a nonterminal into a pair of strings, γ and α, with both terminals and nonterminals in both languages, sub-

ject to the constraint that there is a one-to-one correspondence between nonterminal occurrences on the source and target side. In particular, formally syntax-based models explore hierarchical structures of natural language and utilize only a unified nonterminal symbol X in the grammar,

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle, \quad (1)$$

where $\sim$ is the one-to-one correspondence between X's in γ and α, which is indicated by underscripted co-indices on both sides. For example, some English-to-Chinese production rules can be represented as follows:

$$\begin{aligned} X \rightarrow \langle & X_1 \text{enjoy reading} X_2, \\ & X_1 \text{xihuan(}\textit{enjoy}\text{) yuedu(}\textit{reading}\text{)} X_2 \rangle \\ X \rightarrow \langle & X_1 \text{enjoy reading} X_2, \\ & X_1 \text{xihuan(}\textit{enjoy}\text{)} X_2 \text{yuedu(}\textit{reading}\text{)} \rangle \end{aligned} \quad (2)$$

The set of rules, denoted as $\mathcal{R}$, are automatically extracted from sentence-aligned parallel corpus (Chiang, 2007). First, bidirectional word-level alignment is carried out on the parallel corpus running GIZA++ (Och and Ney, 2000). Based on the resulting Viterbi alignments A_{e2f} and A_{f2e}, the union, $A_U = A_{e2f} \cup A_{f2e}$, is taken as the symmetrized word-level alignment. Next, bilingual phrase pairs consistent with word alignments are extracted from A_U (Och and Ney, 2004). Specifically, any pair of consecutive sequences of words below a maximum length M is considered to be a phrase pair if its component words are aligned only within the phrase pair and not to any words outside. The resulting bilingual phrase pair inventory is denoted as $\mathcal{BP}$. Each phrase pair PP $\in \mathcal{BP}$ is represented as a production rule $X \rightarrow \langle f_i^j, e_k^l \rangle$, which we refer to as *phrasal rules*. The SCFG rule set encloses all phrase pairs, i.e., $\mathcal{BP} \subset \mathcal{R}$. Next, we loop through each phrase pair PP and generalize the sub-phrase pair contained in PP, denoted as SP_e and SP_f subject to $\text{SP} = (\text{SP}_f, \text{SP}_e) \in \mathcal{BP}$, with co-indexed nonterminal symbols. We thereby obtain a new rule.

We limit the number of nonterminals in each rule no more than two, thus ensuring the rank of SCFG is two. To reduce rule size and spurious ambiguity, we apply constraints described in (Chiang, 2007). In addition, we require that the sub-phrases being abstracted by correspondent nonterminals have to be aligned together in the original phrase pair, which significantly reduces the number of rules. We will hereafter refer to rules with nonterminal symbols as *abstract rules* to distinguish them from phrasal rules. Finally, an implicit *glue* rule is embedded with decoder to allow for translations that can be achieved by sequentially linking sub-translations generated chunk-by-chunk:

$$X \rightarrow \langle X_1 X_2, X_1 X_2 \rangle. \quad (3)$$

That is, X is also our sentence start symbol.

During such a rule extraction procedure, we note that there is a many-to-many mapping between phrase pairs (contiguous word sequences without nonterminals) and derived rules (a mixed combination of word and nonterminal sequences). In other words, one original phrase pair can induce a number of different rules, and the same rule can also be derived from a number of different phrase pairs.

3.1 Models

All rules in $\mathcal{R}$ are paired with statistical parameters (i.e., weighted SCFG), which combines with other features to form our models using a log-linear framework. Translation using SCFG for an input sentence f is casted as to find the optimal derivation on source and target side (as the grammar is synchronous, the derivations on source and target sides are identical). By "optimal", it indicates that the derivation D maximizes following log-linear models over all possible derivations:

$$\begin{aligned} P(D) \propto & P_{LM}(e)^{\lambda_{LM}} \times \\ & \prod_i \prod_{X \rightarrow <\gamma,\alpha> \in D} \phi_i(X \rightarrow <\gamma, \alpha>)^{\lambda_i}, \end{aligned} \quad (4)$$

where the set of $\phi_i(X \rightarrow <\gamma, \alpha>)$ are features defined over given production rule, and $P_{LM}(e)$ is the language model score on hypothesized output, the λ_i is the feature weight.

Our baseline model follows Chiang's hierarchical model (Chiang, 2007) in conjunction with additional features:

- conditional probabilities in both directions: $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$;
- lexical weights (Koehn et al., 2003) in both directions: $P_w(\gamma|\alpha)$ and $P_w(\alpha|\gamma)$;

- word counts $|e|$;
- rule counts $|D|$;
- target n-gram language model $P_{LM}(e)$;
- glue rule penalty to learn preference of nonterminal rewriting over serial combination through Eq. 3;

Moreover, we propose an additional feature, namely the *abstraction penalty*, to account for the accumulated number of nonterminals applied in D:

- abstraction penalty $exp(-N_a)$, where $N_a = \sum_{X \rightarrow <\gamma,\alpha> \in D} n(\gamma)$

where $n(\gamma)$ is the number of nonterminals in γ. This feature aims to learn the preference among phrasal rules, and abstract rules with one or two nonterminals. This makes our syntax-based model includes a total of nine features.

The training procedure described in (Chiang, 2007) employs heuristics to hypothesize a distribution of possible rules. A count one is assigned to every phrase pair occurrence, which is equally distributed among rules that are derived from this phrase pair. Hypothesizing this distribution as our observations on rule occurrence, relative-frequency estimation is used to obtain $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$.

We note that, however, these parameters are often poorly estimated due to the usage of inaccurate heuristics, which is the major problem that we alleviate in Sec. 4.

3.2 Decoder

The objective of our syntax-based decoder is to search for the optimal derivation tree D from a forest of trees that can represent the input sentence. The target side is mapped accordingly at each nonterminal node in the tree, and a traverse of these nodes obtains the target translation. Fig. 1 shows an example for chart parsing that produces the translation from the best parse.

Our decoder implements a modified CKY parser in C++ with integrated n-gram language model scoring. During search, chart cells are filled in a bottom-up fashion until a tree rooted from nonterminal is generated that covers the entire input sentence. The dynamic programming item we bookkeep is denoted as $[X, i, j; e_b]$, indicating a sub-tree rooted with X that has covered input from position i to j generating target translation with boundary words e_b. To speed up the decoding, a pruning scheme similar to the cube pruning (Chiang, 2007) is performed during search.

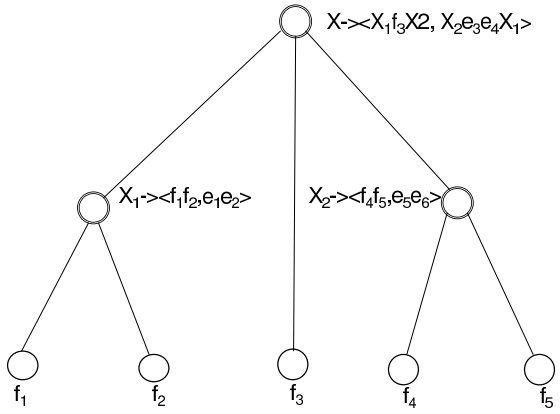

Figure 1: A chart parsing-based decoding on SCFG produces translation from the best parse: $f_1 f_2 f_3 f_4 f_5 \rightarrow e_5 e_6 e_3 e_4 e_1 e_2$.

4 Prior Derivation Models

As mentioned above, decoding searches for the optimal tree on source side to cover the input sentence with respect to given models, as shown in Eq. 4. Among these feature functions, ϕ_i measures how likely the source and hypothesized target sub-trees rooted from same X are paired together through symmetric conditional probabilities (e.g., $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$), and the target language model measures the fluency on target string. It should be noted that, however, the baseline models do not discriminate between different parses on source side when the target side is unknown.

Therefore, if we could obtain some prior distributions for the source side derivations, we can rewrite Eq. 4 as:

$$P(D) \propto P_{LM}(e)^{\lambda_{LM}} \times \prod_{X \rightarrow <\gamma,\alpha> \in D} (\prod_i \phi_i(X \rightarrow <\gamma, \alpha>)^{\lambda_i}) L(X \rightarrow <\gamma, *>)^{\lambda_L}, \quad (5)$$

where $L(\cdot)$ is a feature function defined over a production but only depending on one side of the rules

and asterisk denotes arbitrary symbol sequences on the other side consistent with our grammars [2]. The production of $L(\cdot)$ over all rules observed in a derivation D measures the prior distribution of D. In the baseline model, as a special case, we can see that $L(\cdot)$ is a constant function.

The motivation is straightforward since some derivations should be preferred over others. One may make an analogy between our prior derivation distributions to non-uniform source side segmentation models in phrase-based systems. However, it should be noted that prior derivation models influence not only on phrase choices as what segmentation models do, but also on ordering options due to the nonterminal usage in syntax-based models.

In principle, some quantitative schemes are needed to evaluate the monolingual derivation prior probability. Our scheme links the source side derivation prior probability with the expected ambiguity on target side generation when mapping the source side to target side given the derivation. That is, a given source side derivation is favored if it introduces less ambiguity on target generation compared to others.

Let us revisit the rules in Eq. 2. We notice that the same source side maps into different target orders depending on the *syntactic role* (e.g., NP or PP) of X_2 in the rule. Furthermore, the following are example rules trained from real data (see Sec. 5):

$$X \rightarrow \langle X_1\text{pass}X_2, X_1\text{gei (}\textit{give}\text{)}X_2\rangle \quad (6)$$

$$X \rightarrow \langle X_1\text{pass}X_2, X_1\text{jingguo (}\textit{traverse}\text{)}X_2\rangle \quad (7)$$

$$X \rightarrow \langle X_1\text{pass}X_2, X_1\text{piao (}\textit{ticket}\text{)}X_2\rangle \quad (8)$$

Above three rules cover pretty well for different usages of *pass* in English and its correspondence in Chinese. Typically, applying the rule to inputs such as "my pass *expired*" obtains reasonable translations with baseline models. However, it will fail on inputs like "my pass *to the zoo*" as none of th rules provides a correct translation of X_1X_2piao (*ticket*) when X_2 is a prepositional phrase.

Such linguistic phenomena, among others, indicates that the higher variation of syntax structures the nonterminal embodies, the more translation options on target side needed to account for various syntactic roles on source side. This suggests that our prior derivation models should prefer nonterminals that cover more syntactically homogeneous constituents. Such a model is thus proposed in Sec. 4.1.

The prior derivation model can also be viewed as a smoothing on rule translation probabilities estimated using heuristics, as we mentioned in Sec. 3.1. When there are more translation options, we deem that there are more ambiguity for this rule. In cases where some dominating translation option is overestimated from hypothesized distributions, all translation options of this rule are discounted as they are less favored by prior derivation models.

4.1 Model Syntactic Variations

Each abstract rule is generalized from a set of original relevant phrase pairs by grouping an appropriate set of sub-phrases into a nonterminal symbol, with each sub-phrase linked to a tree list. Therefore, the joined tree lists form a forest for this nonterminal symbol in the rule. For every abstract rule, we define the rule forest to be the set of tree fragments of all sub-phrases abstracted within this rule.

We parse the English side of parallel corpus to obtain a syntactic tree for each English sentence. For each phrase extracted from this sentence, we define the *tree fragment* for this phrase as the minimal set of internal tree whose leaves span exactly over this phrase. As a common practice, we preserve all phrase pairs in $\mathcal{BP}$ including those who are not consistent with parser sub-trees. Therefore, there will be many phrases that cross over syntactic sub-trees, which subsequently produced tree fragments lacking a root. We label those as "incomplete" tree fragments, and introduce a parent node of "INC" on top of them to form a single-rooted subtree. For example, Fig. 2 shows the tree fragments for phrases of "reading books" and "enjoy reading", where the latter is an "incomplete" tree fragment. Moreover, for sentences failed on parsing, we label all phrases extracted from those sentences with a root of "EMPTY".

Subset trees of tree fragments are defined as any sub-graph that contains more than one nodes, with the restriction that entire rule productions must be

[2] In general, we can plug in either $L(X \rightarrow < \gamma, * >)$ or $L(X \rightarrow < *, \alpha >)$ here. For illustration purposes, we assume that the model is on the source side.

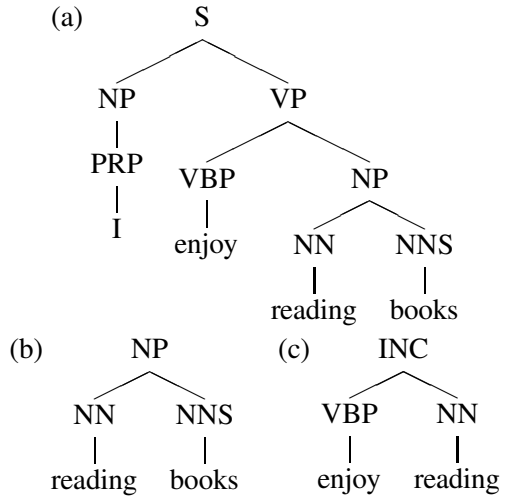

Figure 2: Syntax parsing tree (a) and tree fragments for phrases "reading books" (b) and "enjoy reading" (c).

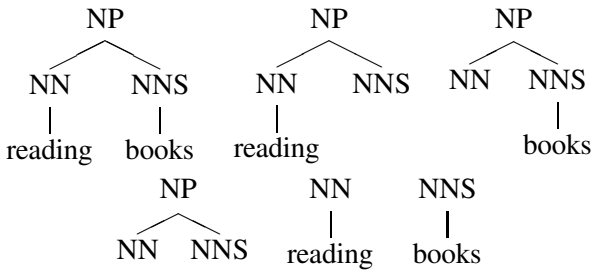

Figure 3: Subset trees of the NP covering "reading books".

included (Collins and Duffy, 2002). Fig. 3 enumerates a list of subset trees for fragment (b) in Fig. 2.

To measure syntactic homogeneity, we define the fragment similarity $K(T_1, T_2)$ as the number of common subset trees between two tree fragments T_1 and T_2. Conceptually, if we enumerate all possible subset trees $1, \ldots, M$, we can represent each tree fragment T as a vector $h(T) = (c_1, \ldots, c_M)$ with each element as the count of occurrences of each subset tree in T. Thus, the similarity can be expressed by the inner products of these two vectors. Note that M will be a huge number for our problem, and thus we need kernel methods presented below to make computation tractable.

4.2 Kernel Methods

Collins and Duffy (Collins and Duffy, 2002) introduced a method employing convolution kernels to measure similarity between two trees in terms of their sub-structures. If we define an indicator function $I_i(n)$ to be 1 if subset tree i is rooted at node n and 0 otherwise, we have:

$$K(T_1, T_2) = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} C(n_1, n_2) \qquad (9)$$

where $C(n_1, n_2) = \sum_i I_i(n_1) I_i(n_2)$ and N_1, N_2 are the set of nodes in the tree fragment T_1 and T_2 respectively. It is noted that $C(n_1, n_2)$ can be computed recursively (Collins and Duffy, 2002):

1. $C(n_1, n_2) = 0$ if the productions at n_1 and n_2 are different;
2. $C(n_1, n_2) = 1$ if the productions at n_1 and n_2 are the same and both are pre-terminals;
3. Otherwise,

$$C(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + C(ch_{n_1}^j, ch_{n_2}^j)) \qquad (10)$$

where $ch_{n_1}^j$ is the jth child of node n_1, $nc(n_1)$ is the number of children at n_1 and $0 < \lambda \leq 1$ is a decay factor to discount the effects of deeper tree structures.

In principle, the computational complexity of Eq. 10 is $\mathcal{O}(|N_1| \times |N_2|)$. However, as noted by (Collins and Duffy, 2002), the worst case is quite uncommon to natural language syntactic trees. More recently, Moschitti (Moschitti, 2006) introduced in details a fast implementation of tree kernels, where a node pair set is first constructed for those associated with same production rules. Our work follows Moschitti's implementation, which runs in linear time on average. We compute the normalized similarity as $K'(T_1, T_2) = \frac{K(T_1,T_2)}{\sqrt{K(T_1,T_1)} \times \sqrt{K(T_2,T_2)}}$ to ensure similarity is normalized between 0 and 1.

4.3 Prior Derivation Cost

First we define the purity of a nonterminal forest (with respect to a given rule) $Pur(X)$ as the average similarity of all tree fragments in the cluster:

$$Pur(X) = \frac{2}{N(N-1)} \sum_j \sum_{i<j} K'(T_i, T_j), \quad (11)$$

where N is number of tree fragments in the forest of X. We now can define the derivation cost $L(X \rightarrow <$

$\gamma, * >)$ for a rule production as:

$$L(X \rightarrow < \gamma, * >) = \\ -\log((\min_{X_1, X_2 \in \gamma}(Pur(X_1), Pur(X_2)))^k), \quad (12)$$

where $k \geq 1$ is the degree of smoothness. Note that the prior derivation cost is set as $L(\cdot) = 0$ by definition for phrasal rules.

Eq. 11 is quadratic complexity with N, however, we note that rules with a large N will typically score poorly on prior derivation models, and thus we can avoid the computation for those by assigning them a large cost. With the fast kernel computation, the training procedure involved with the prior derivation models for the task presented in Sec. 5 is about 5 times slower on a single machine, compared with the training of the baseline system. However, we note that our training procedure can be computed in parallel, and therefore the training speed is not a bottleneck when multiple CPUs are available.

5 Experiments

We perform our experiments on an English-to-Chinese translation task in travel domain. Our training set contains 482017 parallel sentences (with 4.4M words on the English side), which are collected from transcription and human translation of conversations. The vocabulary size is 37K for English and 44K for Chinese after segmentation.

Our evaluation data is a held out data set of 2755 sentences pairs. We extracted every one out of two sentence pairs into the dev-set, and left the remainder as the test-set. We thereby obtained a dev-set of 1378 sentence pairs, and a test-set with 1377 sentence pairs. In both cases, there are about 15K running words on English side. All Chinese sentences in training, dev and test sets are all automatically segmented into words. Minimum-error-rate training (Och, 2003) are conducted on dev-set to optimize feature weights maximizing the BLEU score up to 4-grams, and the obtained feature weights are blindly applied on the test-set. To compare performances excluding tokenization effects, all BLEU scores are optimized (on dev-set) and reported (on test-set) at Chinese character-level.

From training data, we extracted an initial phrase pair set with 3.7M entries for phrases up to 8 words on Chinese side. We trained a 4-gram language model for Chinese at word level, which is shared by all translation systems reported in this paper, using the Chinese side of the parallel corpus that contains around 2M segmented words.

We compare the proposed models with two baselines: a state-of-the-art phrase-based system and a formal syntax-based system as described in Sec. 3. The phrase-based system employs the 3.7M phrase pairs to build the translation model, and it contains a total set of 8 features, most of which are identical to our baseline formal syntax-based model. The difference only lies on that the glue and abstraction penalty are not applicable for phrase-based system. Instead, a lexicalized reordering model is trained from the word-aligned parallel corpus for the phrase-based system. More details about our multiple-graph based phrasal SMT can be found in (Zhou et al., 2006; Zhou et al., 2008). For the baseline syntax-based system, we generated a total of 15M rules and used 9 features.

We chose the Stanford parser (Klein and Manning, 2002) as the English parser in our experiments due to its high accuracy and relatively faster speed. It was trained on the Wall Street Journal section of the Penn Treebank. During the parsing, the input English sentences were tokenized first, in a style consistent with the data in the Penn Treebank.

We sent 482017 English sentences to the parser. There were 1221 long sentences failed, less than 0.3% of the whole set. After the word alignment and phrase extraction on the parallel corpus, we obtained 2.2M unique English phrases. Among them there are about 34K phrases having an empty tree in their corresponding tree lists, due to the failure in parsing. The number of unique tree fragments for English phrases is 2.5M. Out of them there are 750K marked as incomplete. As mentioned previously, each rule covers a set of phrases, with each phrase linked to a tree list. The total number of rules with unique English side is around 8M.

The distribution of the number of rules over the number of corresponding trees is shown in Table 1. We observe that the majority of rules in our model has less than 150 tree fragments. Therefore, considering the quadratic complexity in Eq. 11, we punish the rules with more than 150 unique tree fragments with some floor cluster purity to speed up

Table 1: Distribution of rules over trees

Number of trees	Number of rules
$(0, 10]$	3636766
$(10, 20]$	1556806
$(20, 30]$	989848
$(30, 40]$	916606
$(40, 50]$	488469
$(50, 60]$	270484
$(60, 70]$	198438
$(70, 80]$	86921
$(80, 90]$	58280
$(90, 100]$	29147
$(100, 150]$	437231
> 150	81060

the training. Not surprisingly, the rules with a large number of tree fragments are typically those with few stop words as terminals. For instance, the rule $X \rightarrow < X_1 \mathrm{a} X_2, * >$ comes with more than 100K trees for the X_1.

Table 2: English-to-Chinese BLEU score result on testset (character-based)

Models	BLEU(4-gram)
Phrase-based	42.11
Formally Syntax-based	43.75
Formally Syntax-based with prior derivation	44.51

Translation results are presented in Table 2 with character-based BLEU scores using 2 references. Our baseline formally syntax-based models achieved the BLEU score of 43.75, an absolute improvement of 1.6 point improvement over phrase-based models. The improvement is statistically significant with $p < 0.01$ using the sign-test described by (Collins et al., 2005). Applying the prior derivation model into the syntax-based system, BLEU score is further improved to 44.51, obtained an another absolute improvement of 0.8 point, which is also significantly better than our baseline syntax-based models ($p < 0.05$).

6 Discussion and Summary

We introduced a prior derivation model to enhance formally syntax-based SCFG for translation. Our approach links a prior rule distribution with the syntactic variations of abstracted sub-phrases, which is modeled by distance measuring of linguistically syntax parsing tree fragments using kernel methods. The proposed model has improved translation performance over both phrase-based and formally syntax-based models. Moreover, such a prior distribution can also be used to rank and prune SCFG rules to reduce memory usage for online translation systems based on syntax-based models.

Although the experiments in this paper are conducted for prior derivation models on source side in an English-to-Chinese task, we are interested in applying this to foreign-to-English models as well. As what we pointed out in Sec. 4, target side prior derivation model fits with our framework as well.

7 Acknowledgement

The authors would like to thank Stanley F. Chen and the anonymous reviewers for their helpful comments on this paper.

References

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

Michael Collins and Nigel Duffy. 2002. Convolution kernels for natural language. In *Advances in Neural Information Processing Systems 14*.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proc. of ACL*, pages 531–540.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proc. of HLT/NAACL-04*, Boston, USA, May.

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 *Proc. of ACL*, pages 961–968.

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proc. of ACL*, pages 80–87.

Dan Klein and Christopher D. Manning. 2002. Fast exact inference with a factored model for natural language parsing. In *NIPS*, pages 3–10.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. NAACL/HLT*.

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

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proc. of EACL*.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proc. of ACL*, pages 440–447, Hong Kong, China, October.

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167.

Giorgio Satta and Enoch Peserico. 2005. Some computational complexity results for synchronous context-free grammars. In *Proc. of HLT/EMNLP*, pages 803–810.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23(3):377–403.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proc. of ACL*, pages 523–530.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *Proc. of HLT/NAACL*, pages 256–263.

Bowen Zhou, Stan F. Chen, and Yuqing Gao. 2006. Folsom: A fast and memory-efficient phrase-based approach to statistical machine translation. In *IEEE/ACL Workshop on Spoken Language Technology*.

Bowen Zhou, Rong Zhang, and Yuqing Gao. 2008. Lexicalized reordering in multiple-graph based statistical machine translation. In *Proc. ICASSP*.