Tree-State Based Rule Selection Models for Hierarchical Phrase-Based Machine Translation

Shujian Huang,[1] **Huifeng Sun,**[1] **Chengqi Zhao,**[1] **Jinsong Su,**[2] **Xin-Yu Dai,**[1] **Jiajun Chen**[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, P. R. China
[2]Xiamen University, Xiamen, P. R. China
huangsj@nju.edu.cn, {sunhf,zhaocq}@nlp.nju.edu.cn,
jssu@xmu.edu.cn, {daixinyu,chenjj}@nju.edu.cn

Abstract

Hierarchical phrase-based translation systems (HPBs) perform translation using a synchronous context free grammar which has only one unified non-terminal for every translation rule. While the usage of the unified non-terminal brings freedom to generate translations with almost arbitrary structures, it also takes the risks of generating low-quality translations which has a wrong syntactic structure. In this paper, we propose tree-state models to discriminate the good or bad usage of translation rules based on the syntactic structures of the source sentence. We propose to use statistical models and context dependent features to estimate the probability of each tree state for each translation rule and punish the usage of rules in the translation system which violates their tree states. Experimental results demonstrate that these simple models could bring significant improvements to the translation quality.

1 Introduction

Phrase-based models [Koehn *et al.*, 2003], hierarchical phrase-based models (HPBs) [Chiang, 2005] and syntax-based models [Yamada and Knight, 2001; Liu *et al.*, 2006] are three major types of statistical machine translation models. Besides the decoding constraints and algorithms, the major differences among the three types of systems are the selections of translation equivalences, i.e. translation rules.

Phrase-based translation models use consecutive words as translation rules (called *phrases*). Great improvement is achieved over word-based models due to the ability of resolving word-level reordering inside the phrases. However, despite of the many efforts of modeling reordering operations [Tillmann and Zhang, 2005; Galley and Manning, 2008; Xiong, 2006], phrase-based models are still relatively weak at the phrase-level reordering.

Syntax-based translation models benefit from the reordering of larger translation units using synchronous context free grammars (SCFGs). Several consecutive words, if corresponding to a complete syntactic structure, could be represented by a syntactic non-terminal and reordered as a single element [Yamada and Knight, 2001; Liu *et al.*, 2006]. To ensure the correspondences between translation rules and syntactic structures, syntax-based models usually constrain the translation rules to be consistent with a given monolingual syntactic structure. This constraint severely limits the number of extracted translation rules, which harms the overall translation quality.

HPBs [Chiang, 2005] enjoy the benefit of phrase level reordering by using SCFGs. On the other hand, HPBs use a single unified non-terminal, which represents any consecutive words, bringing in the benefit of phrase-based models. However, while the usage of the unified non-terminal brings freedom to generate translations with arbitrary structures, it also takes the risks of generating low-quality translations which has a wrong syntactic structure.

One way of constraining the extracted rules is by considering the proper boundary of the rules using a discriminative model [Xiong *et al.*, 2010; He *et al.*, 2010; Cui *et al.*, 2010; Zhang *et al.*, 2014]. The probability that a given rule has proper boundaries could be added to the translation model as a feature. These models usually use little syntactic information. And because there is no labeled data for proper rule boundaries, these discriminative models are usually trained with instances extracted using heuristics, which may not be reliable in case of noisy alignment.

Another way of constraining the rules is to use syntactic information, which might be more reliable compared to the above boundary information. However, previous researches mainly emphasize on selecting translation rules that are a complete constituent in a given syntactic structure [Chiang, 2005; Marton and Resnik, 2008; Liu *et al.*, 2011]. The problem with these methods is that, many reasonable rules are actually not a complete constituent. For example (Figure 1), in the translation of ”China faces even more painful choices”, ”China faces” is very reasonable to be a phrase because the singular-plural form agreement should be ensured between the subjective ”China” and the verb ”faces”. However, in syntactic trees, the verb usually first forms a verb phrase (VP) constituent with the objective. In this case, ”China faces” is not a complete constituent and will be punished when it is used in translation.

In this paper, we propose a new angle of viewing the problem. Inspired by the practice of lexicalized reordering models in phrase-based systems [Tillman, 2004], we model the corre-

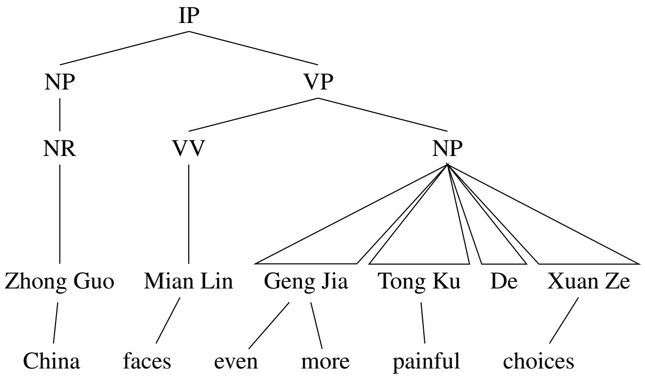

Figure 1: An example of Chinese to English translation with the source side syntactic structure. The inner structure of the objective NP is omitted for simplicity. Chinese characters are represented in the Pinyin form. The translation correspondences are marked with alignment links between Chinese and English words. The phrase ”Zhong Guo Mian Lin - China faces” is not a good phrase from the syntactic perspective, because it crosses the boundary of the VP structure.

spondence between a translation rule and its syntactic structure by three states: *Match*, *Cross*, *Partial*. Our intuition is to use every rule in the most likely tree state as it is extracted from the training data. We firstly perform a maximum likelihood estimation for the probability of every rule be applied in a given tree state. During translation, the probabilities of these three states are used as features to favor the rules that are in the same state as they are originally extracted during the extraction phase.

To incorporate context features, such as boundary words, part-of-speech tags, to improve the probability estimation, we employ statistical models such as the maximum entropy models and the neural network based models. We define forward and backward n-gram features to emphasize the boundary word information, and use distributional word representations to alleviate the data sparsity issue. Experiments show that both context features and the distributional word representations improve the tree state estimation. With the predicted tree state probability, the translation system is able to select suitable translation rules and obtains significant better translation results.

2 Related Work

This paper focuses on the problem of selecting rules based on the source side information. Early approaches are based on firing binary features according to a syntactic tree. Chiang [2005] tried to add a feature into the system to favor phrases that form a constituent of the tree. Marton and Resnik [2008] used separate features for every syntactic label and for rules matching a constituent and rules crossing a constituent, respectively. Our method is based on the probability estimated from training data, and does not have bias towards rule matching or crossing any constituent.

Liu et al. [2011] implemented models using syntactic labels as information source and expanding the crossing case of every syntactic label into three cases: missing left part, missing right part and both. Their approach trains 88 classifiers for different labels, which is not only expensive to implement, but also difficult to integrate using standard minimum error rate training (MERT) [Och, 2003]. Our method uses much simple tree states and could be easily implemented in current systems.

Some approaches make selection decisions based on the boundary of the rule. Xiong et al. [2010] built classifiers on whether the current word is a correct boundary; He et al. [2010] extended the model to include 4 classes: the begin, middle and end of a rule, and single word rule. Zhang et al. [2014] built the classifiers on the rules instead of words. These methods all relies on heuristics in the rule extraction process; while our method gets information from more reliable syntactic analysis results.

The application of a translation rule could be considered from both the source side and the target side [Cui *et al.*, 2010]. These methods could also be applied upon our current work.

3 Definition and notations

3.1 Syntactic Tree

In computational linguistics, context free grammars are used to represent the syntactic structure of sentences. Formally, a syntactic phrase structure grammar G is defined by the 4-tuple $\langle \mathcal{V}, \Sigma, S, \mathcal{P} \rangle$. In the tuple, $\mathcal{V}$ is the set of non-terminals, which represent the syntactic categories, e.g. NP, VP, IP, etc.; Σ is the set of terminals, which is the vocabulary set of a given language; S is the start symbol, which usually represents a complete sentence; P is the set of production rules. Each production rule $p \in \mathcal{P}$ is of the form $V \rightarrow \beta$, where V is the element of $\mathcal{V}$, indicating the syntactic category of p; and β is a string of non-terminals or a string of terminals, indicating how a larger syntactic constituent is composed of smaller ones. These grammars are usually learned automatically from a large set of labeled syntactic trees, for example, PennTreebank[1].

Under a given syntactic grammar G, for any source language sentence $s = s_1, s_2, ..., s_n$, the syntactic parsing tree $T(s)$ is built up of a series of production rules $p_1, p_2, ..., p_k$. Each of these rules p covers a certain part of s, e.g. $s_i, ..., s_j$, denoted as $span(p) = (i, j)$.

3.2 Machine Translation with SCFGs

In machine translation research, synchronous context free grammars (SCFGs) are used to model the process of translation. SCFGs generate strings in both languages in the same time during deduction, thus ensure the correspondence between languages. SCFGs are defined by the 5-tuple $\langle \mathcal{V}, \Sigma_1, \Sigma_2, S, \mathcal{R} \rangle$, where $\mathcal{V}$ is the set of non-terminals; Σ_1 and Σ_2 are terminals in the source and target language, respectively; S is the start symbol. The synchronous production rule $r \in \mathcal{R}$ is in the form $V \rightarrow \langle \gamma, \alpha, \sim \rangle$, where V is the element of $\mathcal{V}$; γ and α are source and target side of the

[1] http://www.cis.upenn.edu/ treebank/

rule, respectively. Each side of the rule is composed of non-terminals in $\mathcal{V}$ or terminals in the corresponding language. $\sim$ is the one-to-one correspondence between the non-terminals in γ and α.

When translating a given sentence s, the translation derivation D is built up of a series of synchronous production rules r, each of which covers source side span $span_s(r)$ and generates a target side span $span_t(r)$. Translation rules, which describe the translation correspondence between a source side string and a target side string, are usually automatically learned from a large parallel corpus and evaluated by some probabilistic models. As stated before, HPBs uses a special kind of SCFG, which has only one unified non-terminal X [Chiang, 2005].

4 Tree-State Models

The proposed tree-state models define three syntactic states for each translation rule when it is applied during translation. We first define the three tree states, then introduce probabilistic models for estimating the probabilities for these states.

4.1 Tree States

When a hierarchical translation rule r is applied during the translation of sentence s. The following three syntactic states of r are defined by considering the relation between the span of r and the span of a production rule p of the syntactic tree T of s.

Match The span of r matches the span of p.

Cross The span of r intersects with the span of p.

Partial The span of r is part of the span of p.

Formally, we define the three tree states as follows:

$$\forall span_s(r) = (i, j), \text{if } \exists p \in T, span(p) = (i', j') \text{ s.t.}$$

$$\begin{cases} i = i' \text{ and } j = j' & \text{r is Match} \\ i' < i \leq j' < j \text{ or } i < i' \leq j < j' & \text{r is Cross} \\ \text{other cases} & \text{r is Partial} \end{cases}$$

4.2 Tree-State Probabilities

Using random variable X to denote the tree state of a hierarchical translation rule r, and x to denote the value of the tree state, we define the tree-state probability of r, $P_{ts}(X = x|r, C(r))$, to be the probability of r being applied in a given tree state x, which depends on the rule r and its context $C(r)$.

Given a translation derivation D, the tree-state probability of D is the product of tree-state probabilities of every r in D (Equation 1). The tree state of r is determined by a syntactic tree of the translated source sentence.

$$P_{ts}(D) = \prod_{r \in D} P_{ts}(X = x|r, C(r)) \tag{1}$$

$P_{ts}(D)$ gives a probability of all hierarchical translation rules in D be applied in their current tree states. Thus it could be used as an estimation of how well the current translation agrees with the parallel data, where all the translation rules are extracted. We add $P_{ts}(D)$ as an additional feature to discriminate different translations in a HPB system.

5 Probability Estimation

Because r is extracted from a large set of parallel sentences, P_{ts} could be estimated according to the context and tree state of r in original sentences. We propose to use the following methods for the probability estimation.

5.1 Maximum Likelihood Estimation

The most fundamental method for estimating the tree state probabilities is Maximum Likelihood Estimation (MLE). The occurrence counts of the tree state and context for each r could be recorded during the rule extraction phase and used for estimation (Equation 2).

$$P_{ts}(X = x|r, C(r)) = \frac{count(x, r, C(r))}{\sum_x count(x, r, C(r))} \tag{2}$$

In practice, because the context of each rule is too sparse to enumerate, our MLE omits the context part and only accumulate the count by the rule itself (Equation 3).

$$P_{ts}(X = x|r) = \frac{count(x, r)}{\sum_x count(x, r)} \tag{3}$$

The process of MLE is straightforward as in the estimation of lexicalized reordering models [Tillman, 2004].

5.2 Context Dependent Estimations

As the parameter space is too sparse, especially when the context information is considered, we seek for feature-based estimation methods.

We start with a series of binary indicator features of word and n-grams as the basic representation of the rule. Because the order of words is crucial for discriminating different syntactic structures, we design position-based n-gram features which indicate the n-grams occur in different positions of the rule (inspired by He et al. [2010]). We notice that previous position-based features append absolute positions to n-grams inside the rule, which may result in unnecessary sparseness. For example, when the same word, say w, occurring at the end of rules with different lengths, standard n-grams with positions will append different position information depending on the length of the rule. However, the most important information to be identified is actually the same for these rules, which is that w occurs at the end.

As an alternative, we describe the n-gram information in a special manner which does not depend on the absolute position of the n-gram to the beginning of the rule, but depends on the relative position to either one of the rule boundaries. Our n-gram feature starts from a given position and goes towards the center of the rule. When the starting position is at the left part of the rule, we append a position according to the left boundary of the rule. This is the same with previous position-based n-grams, which we call *forward n-grams*. However, when the starting position is at the right part of the rule, we append the position according to the right boundary of the rule instead of the left one. Reversed from the order of the text, each n-gram is built from the starting position to the center of the rule (*backward n-grams*).

We could also use n-grams to describe the boundary and context of each rule. So an n-gram may start from outside of the rule and go across the rule boundary. But if the n-gram goes across the other boundary of the rule, we mark the rest words as out-of-rule (#OOR#). For example, for a forward 4-gram starting at two words before a single word translation rule (w_i), the feature is f-bigram-i-2:$w_{i-2}\_w_{i-1}\_w_i\_$#OOR#.

We also use the Part-of-Speech (POS) n-grams as a backoff for word n-grams, in order to cover the rare word cases. These n-grams describe the context before the rule, the boundary information of the rule and the context after the rule, etc. Because the tree state information exists only on the source side of the rule, our features are all defined on the source side.

The detailed description of features for the rule r is listed below, assuming $span_s(r) = (i, j)$.

Forward n-grams N-grams starting at the position i-3, i-2, i-1 and i, respectively. Each n-gram consists of the starting word and its next n-1 words.

Backward n-grams N-grams starting at the position j, j+1, j+2 and j+3, respectively. Each n-gram consists of the starting word and its previous n-1 words.

POS n-grams POS n-grams obtained in forward and backward order.

To generate a set of training instances that are feasible for the training of discriminative models, we perform a random sampling during the rule extraction phase on all possible rules to sample a given portion of them. The random sampling is expected to get a tree-state distribution similar with the MLE.

5.3 Maximum Entropy Models

One common practice of modeling problems with a large set of binary indicator features is using the maximum entropy models [Berger *et al.*, 1996]. The probability of each rule in a given tree state is defined in Equation 4.

$$\begin{aligned} P_{ts}(X = x|r, C(r)) &= p_{\lambda_1^M}(x|r, C(r)) \\ &= \frac{exp[\sum_{m=1}^{M} \lambda_m h_m(x, r, C(r))]}{\sum_x exp[\sum_{m=1}^{M} \lambda_m h_m(x, r, C(r))]} \end{aligned} \quad (4)$$

Where h_m is the m^{th} indicator feature and λ_m is the corresponding weight. The model could be efficiently trained by L-BFGS algorithms [Malouf, 2002].

Comparing to the MLE method which performs estimation on the count of the whole rule and context, feature-based models, such as the maximum entropy models, decompose the whole event into a series of binary features. Because the single features occur much more frequently than the whole event, the data sparseness is alleviated. As a result, the model gives a more robust estimation, especially for rare rules.

5.4 Neural Network based Models

Word embedding is an automatically learned mapping from words into low-dimension vectors based on the context where these words occur in. These vectors are demonstrated effective in evaluating the semantic relations between words. Similar words or words with similar meanings will be mapped to vectors that are close to each other [Mikolov *et al.*, 2013]. In our practice, as an improvement from using two different n-grams representing similar words, we could use two embedding vectors which may have close values. In this way, similar words could have similar effects in determining the tree state, further alleviating the data sparseness problem.

Data	Usage	Sentences
LDC	TM train	8,396,924
Gigaword	LM train	14,684,074
MT03	dev	919
MT04	test1	1,789
MT05	test2	1,083

Table 1: Experimental data and statistics.

To better suit the architecture that learns the word embedding, we use neural network based models which take the embedding n-grams of the rule and context as the input and estimate the likelihood of a given tree state. The input is the concatenation of the embedding of the n-grams, $\vec{e}$. The output $\vec{p}$ is the vector of probabilities for each tree state, which is calculated by Equation 5-7.

$$\vec{p} = f_2(W_2\vec{h} + \vec{b}_2) \quad (5)$$

$$\vec{h} = f_1(W_1\vec{e} + \vec{b}_1) \quad (6)$$

$$f_1(z) = \frac{1}{1 + e^{-z}}, f_2(z) = \frac{e^{z_m}}{\sum_k e^{z_k}} \quad (7)$$

There are a weight matrix W, a bias vector $\vec{b}$ and an activation function f between the input and hidden layers, and between the hidden and output layers, respectively. We denote them by subscripts, such as W_1 and W_2. For the hidden layer and the output layer we use sigmoid and softmax function as the activation function, respectively.

We minimize the cross entropy loss between the output node and the actual tree state of each instance. The training of the neural networks could be performed by the standard backpropagation method [Rumelhart *et al.*, 1988] and the dropout tricks [Srivastava *et al.*, 2014].

Conditions	Match	Partial	Cross	Total Rules
len3	27.2%	35.6%	37.2%	0.58m
len10	18.7%	22.8%	58.5%	1.26m

Table 2: Statistics of the tree states extracted from training data. (Total rules are the numbers in millions.)

6 Experiments and Results

6.1 Data and Preparation

We conduct experiments on a large scale machine translation task. The translation model (TM) of the system is trained on parallel sentences from LDC, including LDC2002E18, LDC2003E14, LDC2004E12, LDC2004T08, LDC2005T10, LDC2007T09, which consists of 8.3 million of sentence pairs. We train a language model (LM) from monolingual data which includes Xinhua portion of Gigaword corpus. We use multi-reference data MT03 as the development data (dev), MT04 and MT05 as test data. These data are mainly in the same genre, avoiding the extra consideration of domain adaptation.

The Chinese side of the corpora is word segmented using ICTCLAS[2]. We use the Berkeley parser [Petrov and Klein, 2007] to obtain the POS tag sequence and the syntactic parse tree for each Chinese sentence. The translation rules for the HPB model are extracted from the parallel sentence pairs with the alignment consistent constraint [Chiang, 2005]. With the syntactic tree generated by Berkeley parser, the tree state of each rule could be determined and used to train the probabilistic models.

6.2 Statistics of the Tree States

We perform a random rule sampling by the ratio 0.001 on all translation rules extracted from the parallel sentences. We collect the tree state statistics in two conditions with the maximum length of the source side rules to be 3 and 10, which correspond to the maximum length of phrases and hierarchical phrases in a HPB system, respectively (Table 2).

As shown in Table 2, only a small portion (27.2%) of the extracted phrases match a complete syntactic structure. When considering hierarchical phrases, the ratio goes down to 18.7%. If the translation is restricted to using only *Match* rules, much fewer rules could be used and that would result in a drop in the translation performance [Chiang, 2005].

The results also show that when the length of the source side rule increases, the distribution of different tree states changes dramatically. Over 58.5% of the rules are in the state *Cross* when the maximum length is 10; while for length 3, the ratio is only 37.2%. This explains why Zhang et al. [2014] uses separate classifiers for rules of different lengths. In our experiments, instead of training one classifier for each specific length, we train two classifiers for phrases and hierarchical phrases, respectively.

6.3 Probability Estimation

We use the sampled rules with their tree states as the training instances for the probability estimation. We run the same rule extraction process on the source and the first reference of the MT03 data, and use the extracted rule and their tree states as evaluation data for the probabilistic tree state models. The estimation performance is evaluated by the classification accuracy of instances on the evaluation data.

We use public available tools to train the maximum entropy model[3] and the neural network based models[4]. For the training of maximum entropy model, we filter features that occur less than 20 times. We set the maximum iteration number of the L-BFGS algorithm to 150.

For training the neural network based models, we compare networks with one and two hidden layers. We set the number of hidden nodes in first hidden layer to be 512, and the number of hidden nodes in the second hidden layer to be 128. We perform mini-batched back-propagation training for 150 iterations, with a mini-batch size 1024 and a dropout rate 0.4.

We first use the maximum entropy models to compare several different feature settings. As a direct variation of the MLE, our first maximum entropy model uses only n-gram features from the source side of the rule. The second model uses the n-gram features from both the rule and the context, augmented with positions (the forward and backward n-grams). The third model uses POS n-grams along with previous context n-grams. We could see from Table 3 that the n-grams with context and position information performs much better than rule n-grams in both settings. The results indicate that the tree state is determined not only by n-grams inside the rule itself, but also by the context around the rule. Also, using POS n-grams further improves the performance.

Our second experiment compares the performances of the maximum entropy models and the neural network based models. As shown in the Table 4, neural network based models show steady improvements over maximum entropy in accuracy. These improvements demonstrate the effectiveness of distributional representations compared with the original n-grams. In the comparison between neural network models with different hidden layers, we choose the one with higher evaluation accuracy (*NN2*) for the following experiments.

6.4 Machine Translation Experiments

We perform machine translation with the proposed probabilistic tree state models. Our translation system is an inhouse implementation of the hierarchical phrase-based translation system [Chiang, 2005]. We set the beam size to 20. We train a 5-gram language model on the monolingual data with MKN smoothing [Chen and Goodman, 1996]. The translation quality is evaluated use 4-gram case-insensitive BLEU. Significant tests are performed using bootstrap re-sampling implemented by Clark et al. [2011].

Our baseline system is a basic HPB system. We also compare our methods with three related researches. The first one is the rule boundary method from Zhang et al. [2014], which does not use syntactic information in deciding the boundary (denoted as *RB*). The second one is a simple count based model, which fires an indicator feature whenever a translation rule is in a given tree state (denoted as *Count*). The third one is the well-known work of Marton and Resnik [2008], denoted as *XP*. In our experiments, the best performed feature in *XP* is XP-Cross, which fires an indicator feature whenever a translation rule is in a *Cross* state and intersects with one of the following syntactic labels: IP, NP, VP, QP, CP, PP, ADJP, ADVP, LCP, DNP.

As we can see from Table 5, using extra information to assist the *HPB* systems always improve the translation performance. The *RB* system gets a relatively small improvement (+0.4 BLEU) compared to other systems because it uses no syntactic information. Both the *Count* and *XP* system use binary indicators for the tree states and achieve moderate improvements (+0.5 and +0.6 BLEU). With our proposed probabilistic tree-state models (*MLE*, *ME*, *NN2*), the translation quality could be further improved. This is because each rule now gets an accurate probability estimation instead of a binary indicator. Compared to *MLE* (+0.6 BLEU), context dependent methods achieve comparable or even higher translation quality because they build more accurate probabilistic

[2] http://ictclas.nlpir.org/

[3] https://github.com/lzhang10/maxent

[4] https://cntk.codeplex.com/

features	train-len3(%)	eval-len3(%)	train-len10(%)	eval-len10 (%)
rule n-grams	71.60	68.83	78.73	76.40
context n-grams	88.39	85.53	83.09	81.48
+POS n-grams	**91.16**	**90.54**	**85.06**	**84.55**

Table 3: Classification accuracy of different feature groups of the maximum entropy model on the training and dev data.

models	train-len3(%)	eval-len3(%)	train-len10(%)	eval-len10 (%)
ME	91.16	90.54	85.06	84.55
NN1	**97.40**	90.89	**95.68**	88.42
NN2	96.94	**91.29**	95.07	**88.86**

Table 4: Classification accuracy of different models on the training and evaluation data. NN1 is a neural network model with a single hidden layer of 512 nodes. NN2 is a neural network model with two hidden layer of 512 and 128 nodes, respectively.

Systems	Dev	Test1	Test2	TestAverage
HPB	34.2	34.2	35.2	34.7(-)
RB	34.3 †	34.6	35.5	35.1(+0.4)
Count	34.5 †	34.7 ‡	35.7 †	35.2(+0.5)
XP	**34.7** †	34.8 ‡	35.7 †	35.3(+0.6)
MLE	**34.7** †	35.0 ‡	35.6 †	35.3(+0.6)
ME	34.5	34.8 ‡	35.7 †	35.3(+0.6)
NN2	**34.7**	34.9 ‡	36.1 ‡	35.5(+0.8)
XP+NN2	**34.7** †	**35.1** ‡§	**36.2** ‡§	**35.7**(+**1.0**)

Table 5: BLEU4 in percentage of different systems. † and ‡ mark results that are significant better than the baseline system (HPB) with $p < 0.05$ and $p < 0.01$, respectively. § marks results that are significant better than the XP system with $p < 0.01$.

models that takes the context into account. The neural network based models (*NN2*) brings a significant improvement of +0.8 BLEU, which is the biggest improvement in all single method systems. Moreover, combining the XP-Cross feature and our neural network based feature (*XP+NN2*) could further improves the translation quality (+1.0 BLEU in all), which is significant compared to both the *HPB* baseline and the *XP* system.

6.5 Translation Analysis

We analyze the differences in translations with and without the tree-state probability models. For each selected system, we collect the rules used in generating the final translation for each sentence, and analyze their tree states.

As in Table 6, compared to the baseline *HPB* system, the *XP* system improves the translation quality by encouraging the use of rules that match a syntactic structure and punishing those cross a syntactic structure. As a result, *XP* system uses much more rules in *Match* state (29.8% v.s. 25.2%) and much less rules in *Cross* state (29.3% v.s. 36.7%). On contrast, the *NN2* system also improves the translation quality, but does not show any preference to the *Match* state rules (25.1% v.s. 25.2%). The improvement comes from selecting the translation rules that have the consistent tree state as they are extracted from the training data. This explains why combining these two features achieves an further improvement.

Systems	*Match*	*Partial*	*Cross*
HPB	7.6 (25.2%)	11.6 (38.1%)	10.8 (36.7%)
XP	7.3 (29.8%)	10.1 (40.9%)	7.2 (29.3%)
NN2	5.6 (25.1%)	8.0 (36.0%)	8.7 (38.9%)

Table 6: Comparison of the number (and percentage) of rules per sentence in different tree state during generating the translation.

7 Conclusion

In this paper, we discuss the relation between the source side syntactic parse tree and the machine translation system using synchronous context free grammars. Instead of using the syntactic structure as constraints to the rules, we propose to use probabilistic models to estimate the probability of the tree state for each translation rule. Instead of encouraging the use of syntactic complete rules, we encourage the use of rules that are consistent with the tree states they are extracted.

We propose features and models to estimate the tree state probability of rules. These probabilities could be used to enhance a state-of-the-art machine translations system. Experiments shows that the translation quality on multiple test sets in a large scale machine translation task is significantly improved by using our method.

Although our implementation is under the hierarchical phrase based translation system, our approach should also be applicable to the phrase-based system as well. In the future, potential gains could be achieved by improving the probability estimation using deep structure neural networks. It is also interesting to investigate tree states under different syntactic categories and their influences to the translation quality.

Acknowledgments

We thank anonymous reviewers for their valuable comments. This work is supported by the National Natural Science Foundation of China (No. 61300158, 61472183), the Jiangsu Provincial Research Foundation for Basic Research (No. BK20130580). Xin-Yu DAI is the corresponding author.

References

[Berger *et al.*, 1996] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, March 1996.

[Chen and Goodman, 1996] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[Chiang, 2005] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *annual meeting of the Association for Computational Linguistics*, 2005.

[Clark *et al.*, 2011] Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of ACL 2011-Short Papers*, HLT '11, pages 176–181, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[Cui *et al.*, 2010] Lei Cui, Dongdong Zhang, Mu Li, Ming Zhou, and Tiejun Zhao. A joint rule selection model for hierarchical phrase-based translation. In *Proceedings of ACL 2010-Short Papers*, pages 6–11, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[Galley and Manning, 2008] Michel Galley and Christopher D. Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the EMNLP 2008*, pages 848–856, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[He *et al.*, 2010] Zhongjun He, Yao Meng, and Hao Yu. Learning phrase boundaries for hierarchical phrase-based translation. In *Coling 2010: Posters*, pages 383–390, Beijing, China, August 2010.

[Koehn *et al.*, 2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of HLT-NAACL2003*, 2003.

[Liu *et al.*, 2006] Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 44th Annual Meeting of the Association of Computational Linguistics*. The Association for Computer Linguistics, 2006.

[Liu *et al.*, 2011] Lemao Liu, Tiejun Zhao, Chao Wang, and Hailong Cao. A unified and discriminative soft syntactic constraint model for hierarchical phrase-based translation. In *MT-Summit*, 2011.

[Malouf, 2002] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[Marton and Resnik, 2008] Yuval Marton and Philip Resnik. Soft syntactic constraints for hierarchical phrased-based translation. In *Proceedings of ACL-08: HLT*, pages 1003–1011, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Och, 2003] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*, pages 160–167, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[Petrov and Klein, 2007] Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *Proceedings of HLT-NAACL*, pages 404–411, Rochester, New York, April 2007. Association for Computational Linguistics.

[Rumelhart *et al.*, 1988] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[Tillman, 2004] Christoph Tillman. A unigram orientation model for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[Tillmann and Zhang, 2005] Christoph Tillmann and Tong Zhang. A localized prediction model for statistical machine translation. In *Proceedings of ACL 2005*, pages 557–564, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[Xiong *et al.*, 2010] Deyi Xiong, Min Zhang, and Haizhou Li. Learning translation boundaries for phrase-based decoding. In *Proceedings of NAACL 2010*, pages 136–144, Los Angeles, California, June 2010. Association for Computational Linguistics.

[Xiong, 2006] Deyi Xiong. Maximum entropy based phrase reordering model for statistical machine translation. In *In Proceedings of COLING-ACL*, pages 521–528, 2006.

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

[Zhang *et al.*, 2014] Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, and Hai Zhao. Learning hierarchical translation spans. In *Proceedings of EMNLP 2014*, pages 183–188, Doha, Qatar, October 2014. Association for Computational Linguistics.