Syntax-Based Translation With Bilingually Lexicalized Synchronous Tree Substitution Grammars

Jiajun Zhang, Feifei Zhai, and Chengqing Zong

***Abstract*—Syntax-based models can significantly improve the translation performance due to their grammatical modeling on one or both language side(s). However, the translation rules such as the non-lexical rule "$\mathrm{VP} \rightarrow (\mathrm{x}_0\mathrm{x}_1, \mathrm{VP} : \mathrm{x}_1\mathrm{PP} : \mathrm{x}_0)$" in string-to-tree models do not consider any lexicalized information on the source or target side. The rule is so generalized that any subtree rooted at VP can substitute for the nonterminal $\mathrm{VP} : \mathrm{x}_1$. Because rules containing nonterminals are frequently used when generating the target-side tree structures, there is a risk that rules of this type will potentially be severely misused in decoding due to a lack of lexicalization guidance. In this article, inspired by lexicalized PCFG, which is widely used in monolingual parsing, we propose to upgrade the STSG (synchronous tree substitution grammars)-based syntax translation model with bilingually lexicalized STSG. Using the string-to-tree translation model as a case study, we present generative and discriminative models to integrate lexicalized STSG into the translation model. Both small- and large-scale experiments on Chinese-to-English translation demonstrate that the proposed lexicalized STSG can provide superior rule selection in decoding and substantially improve the translation quality.**

***Index Terms*—Bilingually lexicalized synchronous tree substitution grammars, discriminative model, generative model, syntax-based statistical machine translation.**

I. Introduction

In RECENT years, interest in the syntax-based translation models has flourished, and these models have led to encouraging progress in the improvement of translation quality [2], [4], [8], [9], [13], [15], [17], [21], [29], [35], [38], [39]. According to [7] and [14], translations using syntax-based models can be cast as a parsing problem. Depending on whether the input is a string or a parse tree, we divide these models into two categories: tree-based parsing models and string-based parsing models.

Manuscript received September 07, 2012; revised December 13, 2012 and March 11, 2013; accepted March 17, 2013. Date of publication March 28, 2013; date of current version April 25, 2013. This work was supported by the Natural Science Foundation of Chinaunder Grant 60975053 and the HiTech Research and Development Program ("863" Program) of Chinaunder Grants 2011AA01A207 and 2012AA011102. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gokhan Tur.

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation Chinese Academy of Sciences. Beijing 100190, China (e-mail: jjzhang@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2013.2255283

Tree-based parsing models include tree-to-string models [10], [15] and tree-to-tree models [7], [13], [35]. Both of these two types of models are based on synchronous tree substitution grammars (STSG). Given the syntactic tree of a source sentence, the tree-based parsing algorithm traverses each node in a top-down manner and identifies all of the translation rules that match the local subtree rooted at the node. Using the matched translation rules, the algorithm either constructs a target tree (tree-to-tree) or directly generates the best target string (tree-to-string).

String-based parsing models include string-to-string models [2] and string-to-tree models [8], [16], [22]. Chiang's string-to-string model (the hierarchical phrase-based model) is constructed based on a degraded synchronous context-free grammar (SCFG) without any linguistic information. The string-to-tree model is a state-of-the-art syntax-based translation model designed to explicitly model the target grammar. This translation model is typical of the STSG-based models [5], [8], [24]. We will use the model to illustrate our ideas throughout this paper.

In string-to-tree translation models, an STSG rule consists of two right-hand sides known as the source-hand and target-hand sides. Traditionally, both sides must follow tree substitution grammar (TSG) rules. However, in string-to-tree translation rules, the target side follows a TSG rule, while the source side follows a CFG rule. Because CFG rules can be understood as simplified TSG rules (following Xiao *et al.* [24]), we use the terms *STSG rule* and *STSG* to denote the translation rules and grammar respectively in the string-to-tree model. In this model, the translation problem is similar to a monolingual parsing problem.

Fig. 1 provides a comparison between monolingual parsing and string-to-tree translation. Both methods convert a string into a tree structure using grammar rules. The difference is that monolingual parsing applies PCFG rules to the conversion of an English string into an English parse tree, whereas the string-to-tree translation model parses the Chinese string using the source-side of the STSG rules and synchronously generates an English tree using the target-side of the STSG rules. In the monolingual parsing community, the PCFG model is often criticized for its lack of lexicalization[1] in expressing syntactic preferences for lexical words, especially when constructing the high-level tree nodes. To overcome this deficiency, lexicalized

[1]Here, *lexicalization* means considering lexicon information when building the tree nodes.

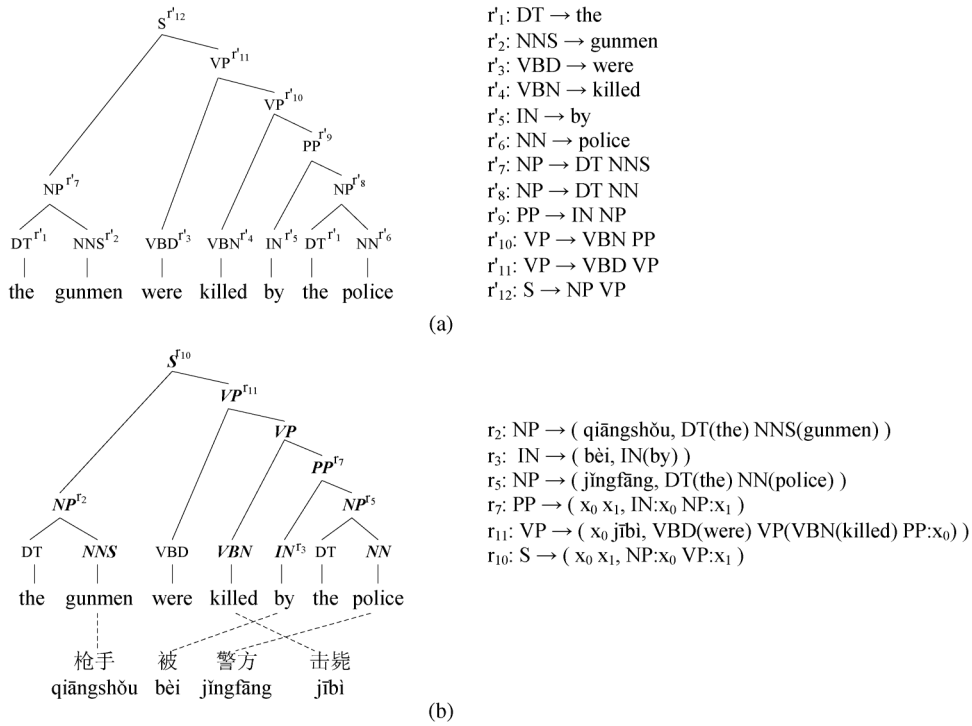

Fig. 1. (a) An example in which an English sentence is parsed into a tree structure with 12 PCFG rules; (b) an instance in which a Chinese sentence (both Chinese characters and Chinese Pinyin are provided, and note that we will use **Chinese Pinyin** throughout the paper) is converted into an English tree using 6 STSG rules. The symbol r_i to the upper right of a node indicates that this node is constructed using rule r_i.

PCFG models such as the Collins' Model 1–3 [6] have been proposed and achieve the state-of-the-art monolingual parsing performance.

Similar to the monolingual PCFG parsing models, the STSG model used in the string-to-tree translation also faces the problem of lacking lexicalization. For example, rule r_{11} in Fig. 1(b) specifies that if the substring x_0 preceding the Chinese word $jībì(kill)$ can be translated into an English prepositional phrase $PP : x_0$, then we can use this rule to translate the Chinese string $x_0\ jībì(kill)$ into the English verb phrase *were killed* $PP : x_0$. Note that it is correct to use this rule only when the Chinese word that is translated into the English preposition indicates the passive voice ($bèi(by)$ in Fig. 1(b), for instance). Otherwise, it is incorrect to apply this rule. For instance, in the string $jǐngfāng(police)yú(in)língchén(morning)$ $jībì(kill)$ $qiāngshǒu(gunmen)$, the Chinese substring $yú(in)\ língchén(morning)$ preceding $jībì(kill)$ can be translated into an English prepositional phrase $PP : x_0$, but the Chinese word $yú(in)$ indicates the active voice. Without considering lexicalization (such as which lexical word generates the English preposition), rule r_{11} is used, and the following poor translation may result: *the police were killed in the morning gunmen*. This detailed example illustrates the importance of lexicalized information in STSG rule selection in the string-to-tree translation.

Inspired by the idea of lexicalized PCFG, we focus on introducing bilingually lexicalized STSG into the string-to-tree translation models. For this purpose, we propose generative as well as discriminative models. Experiments on both small- and large-scale Chinese-to-English translation demonstrate that our proposed bilingually lexicalized STSG can provide superior rule selection for constructing the target trees and can considerably improve the translation quality.

The main contributions of our paper are as follows:

1) In the STSG-based string-to-tree translation model, we propose to incorporate both source- and target-side lexicalized information into the STSG rules to alleviate the overgeneralization problem of STSG and enable the model to choose appropriate translation rules during decoding.
2) To evaluate bilingually lexicalized STSG more thoroughly, we not only propose a generative model using the adapted Collins' Model 1, but also introduce a discriminative model that can incorporate arbitrary lexicalized features.

The remainder of this article is organized as follows. Section II describes the related work and Section III provides a brief overview of STSG-based string-to-tree translation models and explains in detail why bilingually lexicalized parsing models are needed. In Section IV, we elaborate on the proposed bilingually lexicalized parsing models, in which both generative and discriminative models are introduced. Extensive experiments and their analysis are presented in Section V. Finally, we conclude in Section VI.

II. Related Work

To our knowledge, very few researchers investigated bilingually lexicalized STSG parsing models in the context of statistical machine translation.

Many researchers have focused on the contribution of the lexicalized monolingual parsing models, from two directions. One approach applies the lexicalized monolingual parsing model to syntax-based language modeling [1], [18], [20], [24].

Charniak *et al.* [1] first attempted to improve the grammaticality of the output of a string-to-tree system [28] using a lexicalized monolingual PCFG parsing model. Och *et al.* [18] concentrated on phrase-based models. Both of these studies are based on the Penn Treebank trained PCFG parsing model and used the model for re-ranking. However, they found that the lexicalized monolingual parsing model cannot improve the BLEU score. Post and Gildea [20] demonstrated that tree substitution grammars (TSGs) may be a better choice than context-free grammars for language modeling. Xiao *et al.* [24] studied syntax-based language modeling using the lexicalized monolingual parsing model with TSG. The model parameters were directly learned from the automatically parsed target trees. They reported that their approach can improve the translation quality of the string-to-tree model [8].

The other approach employs the lexicalized monolingual parsing model in parsing the source string in the joint parsing and decoding stages [14]. This approach aims to improve the tree-to-string translation model by constructing parse trees on the source side with the help of the monolingual parsing model and generating translations on the target side simultaneously.

In contrast to these previous studies, our approach does not use the parsing model as a language model, and the model parameter learning process does not depend on the Penn Treebank. Instead, we propose a bilingually lexicalized STSG parsing model and used the model to distinguish good rules from poor ones in the decoding stage. The model parameters can be learned easily from the extracted STSG translation rules. In addition to the generative models, we have also investigated a discriminative model. The algorithm using the lexicalized monolingual TSG as a language model is just a special case of our generative model.

Many researchers are also concerned with the overgeneralization problem of STSG rules. For example, Chiang *et al.* [3] designed many target-side syntax features to improve the string-to-tree translation. Rather than using a rule lexicalization model, binary features such as rule overlap features, bad-rewrite features and insertion features are employed in the algorithm. They argue that certain nonterminals are more reliable than others for the rule overlap feature, and they created a binary feature for each root nonterminal of a specified rule. Regarding the bad-rewrite feature, they observed that the nonterminals in certain non-lexical rules are associated only with specific lexical words, and binary features were created to penalize the use of these rules. The features proposed by Chiang *et al.* are also language-dependent, based on a thorough analysis of the tuning set. In contrast to Chiang *et al.*, we argue that in any language pair, the newly generated nonterminal in the target parse tree depends strongly on the underlying source- and target-side lexical strings. We therefore propose the language-independent generative and discriminative lexicalized STSG models, and regard these models as additional strong features to be integrated into the log-linear model. Moreover, to establish a strong baseline, we also include the discount feature used by Chiang *et al.* [3] in our baseline string-to-tree model.

Other related work focuses on grammar or rule lexicalization. An example of grammar lexicalization is the proposal of Zhang and Gildea [31], [32] to improve the word alignment using lexicalized inversion transduction grammar. In rule lexicalization, the syntax-based translation is inherently lexicalized using dependency structures [21], [22], [25], [27]. In this paper, we concentrate on the phrase structure-based string-to-tree translation models that construct a phrase structure tree on the target side during decoding. Using our proposed bilingually lexicalized parsing models, we not only retain the merits of the phrase structure information in the string-to-tree model but also enrich it with source- and target-side lexicalized knowledge. Other approaches consider lexical information during the rule extraction in syntax-based translation models. However, these approaches are not focused on using the rule lexicalization model to distinguish between competing rules. For example, Wu *et al.* [23] considered function words in forest-to-string rule extraction to alleviate the errors in word alignment between source-side words and target-side function words.

III. The String-to-Tree Translation Model

The string-to-tree translation model aims to render the target output more grammatical by explicitly constructing a parse tree on the target side. The model accepts a source string as an input, searches through all of the possible target trees, and finally identifies the tree with the highest score. This process is performed by recursively applying STSG rules that are extracted and parameterized using a word-aligned, target side-parsed parallel training corpus.

A. STSG Translation Rules

Galley *et al.* [9] proposed the GHKM algorithm for extracting (minimal) STSG rules from a triple (f, e_t, a), where f is the source-language sentence, e_t is a target-language parse tree whose yield e is the translation of f, and a is the set of word alignments between e and f. The minimal string-to-tree rules are extracted in three steps: (1) frontier set computation, (2) fragmentation; and (3) extraction.

The frontier set (**FS**) is the set of frontier nodes that meet the following alignment constraint. The target phrase dominated by the frontier node and its corresponding source phrase must be consistent with the word alignment. The ***bold italic*** nodes in the English parse tree in Fig. 1(b)[2] are all frontiers.

Given **FS**, the graph G composed of the triple (f, e_t, a) is divided into several fragments. Each fragment takes only nodes in **FS** as the root and leaf nodes.

Each fragment forms an STSG translation rule. These rules are extracted through a depth-first traversal of e_t: for each frontier visited, a rule is extracted using the fragmentation rooted at this frontier. The extracted rules are known as *minimal rules* [9]. For example, $\mathrm{r}_1 - \mathrm{r}_{10}$ in Table I are minimal rules. To improve the rule coverage, SPMT models [16] can be employed to obtain the *phrasal rules* that are not covered by the GHKM rules. In addition, the minimal rules that share adjacent tree fragments can be connected together to form composed rules [8]. In Table I, r_{11} is a rule composed by combining r_6 and r_9.

The extracted STSG rules are associated with multiple probabilities, such as the phrasal-like translation probabilities and

[2]To some extent, we abuse the example in Fig. 1(b) in that we use it as both a translation example and a training example.

TABLE I
MINIMAL AND COMPOSED RULES EXTRACTED FROM FIG. 1(b)

r_1: NNS → (qiāngshǒu, NNS(gunmen))
r_2: NP → (qiāngshǒu, DT(the) NNS(gunmen))
r_3: IN → (bèi, IN(by))
r_4: NN → (jǐngfāng, NN(police))
r_5: NP → (jǐngfāng, DT(the) NN(police))
r_6: VBN → (jībì, VBN(killed))
r_7: PP → (x_0 x_1, IN:x_0 NP:x_1)
r_8: VP → (x_0 x_1, VBN:x_1 PP:x_0)
r_9: VP → (x_0 x_1, VBD(were) VP(VBN:x_1 PP:x_0))
r_{10}: S → (x_0 x_1, NP:x_0 VP:x_1)
r_{11}: VP → (x_0 jībì, VBD(were) VP(VBN(killed) PP:x_0))

the root-normalized translation probability. The root-normalized probability $P(rule|root)$ is similar to the rule probability of the PCFG in monolingual parsing. In this sense, the string-to-tree translation already employs a PCFG-like model.

B. Decoding as Parsing

Using the extracted STSG translation rules, we now elaborate on the decoding process in the string-to-tree model based on the example in Fig. 1(b).

The decoding process is usually formalized as a deductive system that performs a bottom-up CKY-style parsing algorithm [2], [8], [14]. Fig. 2 illustrates the deductive steps in detail. First, axiom rules r_2, r_3 and r_5 are employed to deduce one-word translations. Then, inference rules r_7, r_{11} and r_{10} are applied to deduce two-word, three-word and four-word translations. We use inference rule r_{10} for the following analysis. The deductive step can be formalized as follows:

$$\frac{NP_{0,1} : (w_1, e_1) VP_{1,4} : (w_2, e_2)}{S_{0,4} : (w, e_1 e_2)} \tag{1}$$

where $NP_{0,1}$ (in which the subscript denotes the source-side index) is deduced using axiom rule r_2, and $VP_{1,4}$ is deduced using inference rule r_{11}. Here, w_1 and e_1 denote the score and the partial translation, respectively. Nodes NP and VP in rule r_{10} are substituted by newly deduced structures and the resulting score for $S_{0,4}$ is calculated as follows:

$$w = w_1 + w_2 + w_{new} \tag{2}$$

Here w_{new} includes the increased language model score and the scores of the rule-related sub-models:

$$w_{new} = w_{lm} + w_{phr} + w_{root} \tag{3}$$

where w_{phr} denotes the phrasal-like scores, and w_{root} is the PCFG-like parsing model score, which is indicative of whether the generated target tree is well-formed. For the example rule r_{10}, w_{root} is computed as follows (the model weight λ is not explicitly given):

$$w_{root} = \log P\left(S \rightarrow (x_0 x_1, NP : x_0 VP : x_1) | S\right) \tag{4}$$

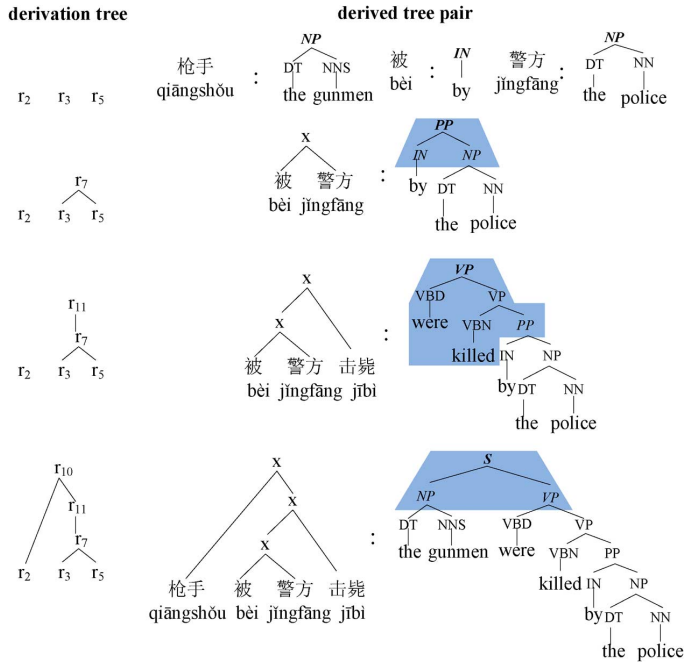

Fig. 2. Illustration of string-to-tree decoding.

From the translation process illustrated in detail in Fig. 2, we can see that when constructing the target-side lower tree nodes, both source- and target-side lexicalized information is used in the application of the axiom rules such as r_2, r_3 and r_5, and these axiom rules are therefore reliable. However, when the target tree is extended from lower to higher levels, the inference rules containing nonterminals are increasingly utilized, and these rules are less reliable as the inside generalized nonterminals are not informed by the lexical evidence. For example, rule r_{10} does not consider any lexicalized information, and the PCFG-like score w_{root} cannot capture the knowledge which represents the preference to the source- and target-side lexicalized information. Rules of this type generally suffer from a lack of reliability.

Based on our analysis, the string-to-tree translation model requires rich lexicalized information for reliable application of the STSG rules during decoding. Inspired by Collins' lexicalized PCFG parsing models, we therefore propose to enhance string-to-tree translation models with bilingually lexicalized STSG.

IV. DECODING WITH BILINGUALLY LEXICALIZED STSG PARSING MODELS

In this section, we first review the necessary background on bilingually lexicalized STSG parsing models, including lexicalized STSG rule extraction and rule conversion. We then propose two parameter estimation methods (a generative model and a discriminative model). Finally, we demonstrate how to apply lexicalized STSG parsing models in decoding.

A. Lexicalized STSG Rule Extraction

Informally, we define a lexicalized STSG rule as an STSG rule in which each nonterminal is associated with source- and target-side lexicalized information.

A question arises regarding the type of lexicalized information that can be used. For the target-side lexicalized training

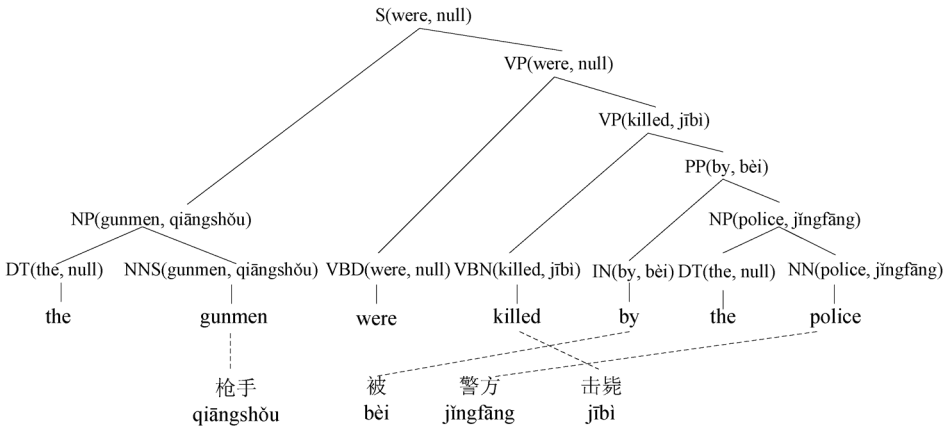

Fig. 3. Lexicalized training example (POS of target headword is not explicitly given).

parse trees, the intuitive idea is to apply the headword information[3] that is used in Collins' lexicalized PCFG parsing models. Before rule extraction, we first use the head finding rules to annotate every interior node of the target-side parse tree with the node's headword and its part-of-speech. For the source-side lexicalized information, we here use a heuristic and associate each node with the source word that is aligned to the target headword of the node. However, we may often encounter many-to-one alignment situations when handling words in the source. In this case, we heuristically choose the source word that is aligned to the target headword of the node with the highest probability[4]. Fig. 3 shows the lexicalized version of the English parse tree in Fig. 1(b).

After annotating the target-side tree nodes with lexicalized information from both sides, the extraction process for the lexicalized STSG rules is the same as that for traditional STSG rules, except that the nonterminals in the lexicalized STSG rules are associated with lexical words. For example, the lexicalized version[5] of rule r_{10} in Table I is as follows:

$$S(were, null) \rightarrow (x_0 x_1, NP(gunmen, qiāngshǒu) : x_0 VP(were, null) : x_1)$$

The height of this STSG rule is one, and the rule is equivalent to an SCFG rule. However, most of STSG rules in string-to-tree models have a height greater than one on the target side (see, e.g., rules r_9 and r_{11} in Table I). In our training data, we found that approximately 86.7% of the STSG rules have heights of two or more. Because the Collins' lexicalized parsing model is based on PCFG rules with heights of one, we must develop an effective method of converting the STSG rules into equivalent SCFG rules of height one to apply the well-studied lexicalized PCFG parsing models.

[3]DeNeefe and Knight [37] also considered headword in machine translation; however their tree-adjoining grammar is more complicated.

[4]Based on our manual analysis of 1000 randomly chosen pairs of target-side headwords and aligned source-side words, we find that the fraction of correctly aligned source-side words is 71.8%. In addition to this heuristic, we can also apply other methods (such as pairwise mutual information and likelihood ratios) to choose the source-side lexical words. We leave this for our future work.

[5]We omit target-side POS tags in the example rule for simplicity.

B. Rule Conversion for Parameterization

Post and Gildea [20], and Xiao *et al.* [24] adopted a 3-step preprocessing approach to transform the training example (target tree, source string and word alignment) to an equivalent structure from which we can extract SCFG rules. This is an effective approach; however, the headword identification process is affected by the deletion of many necessary interior nodes. Alternatively, we can focus directly on the STSG rules.

From the decoding process shown in detail in Fig. 2, we can see that the interior nonterminals (such as VBD, VP and VBN in r_{11}) contribute nothing to the generation of the target translation. Therefore, all of nonterminals of this type can be discarded before decoding. In practice, if a POS tag is among the useless nodes, we still keep it for calculating the Collins' Model rule probability. All interior nonterminals except for the POS tags are therefore removed in each lexicalized STSG rule. For example, the following lexicalized rule is obtained for r_{11}:

$$VP(were, null) \rightarrow (x_0\ j\bar{\imath}b\grave{\imath}, VBD(were, null) VBN(killed, j\bar{\imath}b\grave{\imath}) PP(by, b\grave{e}i) : x_0)$$

Using rules of this kind, we can estimate and predict the Collins' lexicalized PCFG model probabilities. As Collins' model is used only to model the generation of the monolingual grammar rules, we must perform an additional conversion for the parameterization. After attaching the source-side words (which are aligned to the target headwords) to the target-side nonterminals, the rule's source-side will be useless during parameter estimation, and thus we can discard the rule's source-side for parameterization. In the example above, rule will becomes:

$$VP(were, null) \rightarrow VBD(were, null) VBN(killed, j\bar{\imath}b\grave{\imath}) PP(by, b\grave{e}i) : x_0$$

C. A Generative Model

We first apply Collins' Model 1 [6] to model the generation of the simplified lexicalized STSG translation rules.

Collins' Model 1 breaks down the generation of a lexicalized grammar rule into a sequence of sub-steps. More formally, the generation process for the rule $P(h) \rightarrow L_n(l_n) \cdots L_1(l_1) H R_1(r_1) \cdots R_m(r_m)$ is described as follows:

TABLE II
BACKOFF LEVEL FOR EACH SUB-MODEL IN BILINGUALLY LEXICALIZED STSG BACKOFF LEVEL

Backoff Level	$P_h(H\|\cdots)$	$P_l(L_i(l_i)\|\cdots)$ $P_r(R_i(r_i)\|\cdots)$
1	P,h,t	P,H,h,t,Δ
2	P,h_{tgt},t	P,H,h_{tgt},t,Δ
3	P,t	P,H,t,Δ
4	P	P,H,Δ

1) Generate a head constituent label H with probability $P_h(H|P,h)$, where P is the label of the parent node and h is its headword.
2) Generate all of the left and right modifiers, with probabilities $\prod_{i=1\ldots n+1} P_l(L_i(l_i)|P,H,h,t,\Delta)$ and $\prod_{i=1\ldots m+1} P_r(R_i(r_i)|P,H,h,t,\Delta)$. Here L_i (or R_i) is the i-th left (or right) modifier, and l_i (or r_i) denotes the headword of L_i (or R_i). The POS tag of h is denoted by t, and Δ is the distance between the modifier and the head. $L_{n+1}(l_{n+1}) = STOP$ and $R_{m+1}(r_{m+1} = STOP$.

When incorporating Collins' Model 1 into our bilingually lexicalized STSG parsing model, the headword h, l_i and r_i become the target headword and its aligned source word. All of the parameters described above can be estimated using the maximum likelihood method based on the extracted and converted lexicalized STSG rules (in contrast to using Treebank for parameterization in the monolingual parsing). Given a trained model, the probability of a rule can be formalized as follows:

$$\begin{aligned} &P\left(L_{n+1}(l_{n+1})\cdots L_1(l_1)HR_1(r_1)\cdots R_{m+1}(r_{m+1})|P(h)\right) \\ &\quad = P_h(H|P,h)\times\prod_{i=1\ldots n+1} P_l\left(L_i(l_i)|P,H,h,t,\Delta\right) \\ &\qquad \times\prod_{i=1\ldots m+1} P_r\left(R_i(r_i)|P,H,h,t,\Delta\right) \end{aligned} \tag{5}$$

For example, the generative probability of the rule $VP(were,null) \rightarrow VBD(were,null)VBN(killed,j\bar{\imath}b\grave{\imath})PP(by,b\grave{e}i) : x_0$ can be calculated as follows: (the POS tag and distance are omitted for simplicity, but in practice, they are included for model training and prediction):

$$\begin{aligned} &P_h\left(VBD|VP,(were,null)\right) \\ &\quad\times P_l\left(STOP|VP,VBD,(were,null)\right) \\ &\quad\times P_r\left(VBN(killed,j\bar{\imath}b\grave{\imath})|VP,VBD,(were,null)\right) \\ &\quad\times P_r\left(PP(by,b\grave{e}i)|VP,VBD,(were,null)\right) \\ &\quad\times P_r\left(STOP|VP,VBD,(were,null)\right) \end{aligned} \tag{6}$$

It is straightforward to show that this generative lexicalized model is highly specific and tends to suffer from data sparseness. As suggested by Collins [6] for lexicalized PCFG, we therefore design a backoff smoothing mechanism for each sub-model of the proposed bilingually lexicalized STSG in Table II. First, we consider all of the conditions, and then we exclude the condition based on the source-side head information. Levels 3 and 4 are similar to the lexicalized PCFG used in monolingual parsing.

Note that the parsing model including only the target-side lexicalized information is just a backoff of our bilingually lexicalized STSG parsing model. Traditionally, researchers (e.g., [1], [20], [24]) have considered the target-side lexicalized parsing model as an augmented target language model in which source-side lexicalized information cannot be used. Here, we see that the traditional method is a special case of our generative model. In our experiments, we compare the performance of the monolingual lexicalized STSG parsing model (incorporating only the target-side lexicalized information) with the bilingually lexicalized STSG parsing model to figure out whether the lexicalized information from both sides is more helpful.

D. A Discriminative Model

From formula (5), we can see that the generative model utilizes only the headword lexicalized information. A natural question is whether we can incorporate other lexicalized information in addition to the headword.

In addition to the generative model, which models the generation process of a STSG translation rule, we can also consider that the usage preference of the STSG translation rules in decoding can be determined by the lexicalized string pair (target English string (ts) which is spanned by the rule's root node, and its aligned source Chinese string (ss)). We can therefore employ a discriminative model for modeling the conditional distribution:

$$P(P \rightarrow L_n\cdots L_1HR_1\cdots R_m|ts,ss)$$

This model can utilize far more lexicalized information in addition to the headword in the generative model.

To simplify the calculation of the conditional probability, we assume that the nonterminal nodes are conditionally independent of one another and that each nonterminal node in a rule is determined only by the target string spanned by this node and the aligned source string. Moreover, we ignore the structure of the rule (e.g., the order of the nonterminal nodes). The calculation of the conditional probability can then be formalized as follows:

$$\begin{aligned} &P(P \rightarrow L_n\cdots L_1HR_1\cdots R_m|ts,ss) \\ &\quad = P\left(\langle P, L_n\cdots L_1HR_1\cdots R_m\rangle|ts,ss\right) \\ &\quad = P(P|ts_P,ss_P)\times P(H|ts_H,ss_H) \\ &\qquad\times\prod_{i=1\ldots n} P\left(L_i|ts_{L_i},ss_{L_i}\right) \\ &\qquad\times\prod_{i=1\ldots m} P\left(R_i|ts_{R_i},ss_{R_i}\right) \end{aligned} \tag{7}$$

where ts_H denotes the target string, dominated only by the nonterminal node H, and ss_H is the aligned source string corresponding to ts_H. Using formula (7), we can decompose the probability of a rule (given a string pair) into products of probability of each nonterminal in the rule (given its spanned substring pair). Accordingly, we must model the conditional distribution of a nonterminal given a string pair (e.g., $P(H|ts_H,ss_H)$). For example, when applying the rule r_{10} in Fig. 2, $P = S$, $H = VP$ and $L_1 = NP$. For $H = VP$, $ts_{H=VP} = were\ killed\ by\ the\ police$, and

$ss_{H=VP} = bèi\ jǐngfāng\ jībì$. Our task is to predict the tag VP with the bilingual strings.

The problem now becomes a traditional classification problem in which the classes include all of the constituent labels in the Treebank. There are many sophisticated models designed for this problem, such as SVM, neural networks and maximum entropy models. In this article, we formulate the probability using a maximum entropy model:

$$P(H|ts_H, ss_H) = \frac{\exp\left(\sum_i \lambda_i f_i(H, TS_H, SS_H)\right)}{\sum_{H'} \exp\left(\sum_i \lambda_i f_i(H', TS_{H'}, SS_{H'})\right)} \quad (8)$$

where f_i is a binary feature and λ_i is its feature weight. Any informative feature deduced from the string pair (ts_H, ss_H) can be used. Xiong *et al.* [26] and Zollmann and Vogel [36] indicate that the boundary words (the leftmost and rightmost words of a string) are good substitutes for the entire string in phrase reordering and constituent construction. Moreover, the success of Collins' parsing models shows that the headword is also an informative feature. Inspired by their works, we therefore define six features as follows: $f_0 = ts_H$'s leftmost word, $f_1 = ss_H$'s leftmost word, $f_2 = ts_H$'s rightmost word, $f_3 = ss_H$'s rightmost word, $f_4 = ts_H$'s headword, $f_5 = \text{source} - \text{sided word}$ aligned to f_4.

The training examples for the maximum entropy model can easily be extracted prior to the STSG rule extraction process. For each frontier node recognized in the target parse tree, we can derive an example of the form (*node_tag, target_string, source_string*), from which the six representative features mentioned above can be extracted. Consider the $VP(killed, jībì)$ node in Fig. 3 as an example. The extracted training instance is $(VP, killed\ by\ the\ police, bèi\ jǐngfāng\ jībì)$, and the corresponding lexicalized bilingual features are $(killed, jībì, killed, police, bèi, jībì)$. In our large-scale experiment, we totally extract 27,382,983 training examples in which there are 849,952 headwords.

We utilized the toolkit of Zhang [34] to train the maximum entropy model and set the Gaussian prior to $g = 1.0$ to avoid overtraining. We measured the accuracy of the classifier using a held-out data, and we get an accuracy of 76.8% using the six features mentioned above in the large-scale experiment.

E. Applying Lexicalized STSG in the Decoding Stage

Based on the discussion above, we know how to extract the lexicalized rules and estimate their lexicalized probabilities during the training stage. A remaining question is how to apply the lexicalized STSG during decoding. It is important to note that the lexicalized information (e.g., headword, boundary words) is fully connected to the rules when training the lexicalized STSG models; however, the matching rules used during decoding should not be encoded using such lexicalized information. If the rules used in decoding are fully annotated by both source- and target-side lexicalized information, then the resulting rule search space will be extremely sparse.

For example, assumes that we directly apply a rule, such as $PP(by, bèi) \rightarrow (x_0x_1, IN(by, bèi)\ :x_0NP(police, jǐngfāng)\ :\ x_1)$, in the rule matching during decoding. This example rule can be used only when the two subtrees rooted at IN and NP require the headword information $(by, bèi)$ and $(police, jǐngfāng)$, respectively. Clearly, this type of usage is not realistic. A more appropriate method is to employ the original STSG translation rules of the string-to-tree model in the rule matching while generating the bilingually lexicalized information for the rules dynamically during decoding.

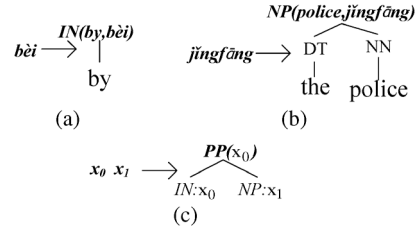

Fig. 4. Examples of rules used during decoding.

For this purpose, we have designed a simple algorithm, which is illustrated through examples in Fig. 4. For the purely lexicalized rules (axiom rules), such as rules (a) and (b), we retain the target-side headwords and aligned source-side words of the root node of the rule (note that the headwords are not combined with the root as matching units). For other rules, such as the non-lexical rule (c), we can simply record the position of origin of the headword information ($PP(x_0)$ means that the headword of PP originates from $IN\ :\ x_0$, and if $IN\ :\ x_0$ is replaced by (a), then (by, $bèi$) passes to the node PP). In this way, the headword information of a rule can be generated dynamically in a bottom-up manner during decoding. For the boundary words used in the discriminative model, we can easily obtain them as we know which source-side span is currently being handled and also know the target-side partial translation after applying an STSG rule.

From the bilingually lexicalized information generated for a given STSG rule, we can calculate the lexicalized STSG probability for this rule using a generative model or discriminative model. This lexicalized STSG probability serves as a strong feature, similar to the n-gram language model, that can guide the decoder to choose the appropriate rules regarding the source- and target-side lexicalized information.

V. Experiments

In our experiments, we incorporate the lexicalized STSG parsing models into the standard log-linear string-to-tree translation model [8], [16], and test the effectiveness of our proposed models.

We first describe the experimental set-up and then we present the experimental results on both small- and large-scale Chinese-to-English translation data sets. For each of these data sets, we wish to determine whether the generative or discriminative model can provide greater improvement in the translation quality and whether the bilingually lexicalized STSG parsing model outperforms the monolingual lexicalized STSG parsing model. In addition to the translation results, we study the influence of the lexicalized STSG parsing models on the rule usage distribution in decoding in detail and investigate the types of rules that are most essential to the lexicalized STSG parsing

TABLE III
EXPERIMENTAL RESULTS FOR THE VARIOUS TRANSLATION SYSTEMS ON TWO TEST SETS. **BOLD** FIGURES INDICATE THAT THE PERFORMANCE IS SIGNIFICANTLY BETTER THAN ***S2T*** ($P < 0.05$). THE VALUES IN THE LAST COLUMN INDICATE THE AVERAGE IMPROVEMENTS OVER JOSHUA AND S2T

Systems	BLEU (%)		
	MT04	MT05	Ave. Gain
Joshua	30.71	27.86	N/A
S2T	33.73	30.25	2.71/N/A
S2T.Gen.Tgt.Lex	34.24	30.58	3.13/0.42
S2T.Gen.SrcTgt.Lex	**34.51**	**30.95**	3.45/0.74
S2T.Dis.SrcTgt.Lex	**34.56**	**30.82**	3.41/0.70

models. We also discuss the relationship between the lexicalized STSG parsing models and the composed rules, which are believed to encode a large portion of the lexicalization. Finally, we analyze the decoding efficiency of the proposed models and present several translation examples.

A. Experimental Set-Up

Our first dataset is the small FBIS bilingual corpus[6] consisting of 236 K sentence pairs. We employed GIZA++ and the *grow-diag-final-and* balance strategy to generate the final symmetric word alignments. We parsed the English side using the Berkeley parser [19]. Given the target-side parsed, word-aligned bilingual corpus, we applied the rule extraction algorithm described in Section III to extract the string-to-tree STSG translation rules. In addition to the GHKM minimal rules [9], we also extracted rules based on SPMT Model 1 [16] with source-side phrases up to length $L = 5$. We also extracted composed rules [8] by combining two or three adjacent minimal rules. Following [30], we binarized the rules to facilitate the language model integration. We trained a 5-gram language model using the target part of the bilingual data and the Xinhua portion of the English Gigaword corpus.

We used the 2003 NIST MT Chinese-to-English test set as the development set and the 2004 and 2005 NIST test sets as our test set. The final translation quality is evaluated in terms of case-insensitive BLEU-4 metric with shortest length penalty, and the statistical significance test is performed using the pairwise resampling approach [11].

B. Experimental Results on the Small-Scale Data

First, we provide brief descriptions of the translation systems used in our experiment.

Joshua: Joshua is a freely available decoder for hierarchical phrase-based SMT [2], [12]. We employ Joshua as a baseline with k (in the k-best outputs) set to be 500 and other settings left at their default values.

S2T: ***S2T*** is our in-house string-to-tree translation system, which is re-implemented following [8], [9] and [16].

S2T.Gen.Tgt.Lex: This string-to-tree translation system integrates the generative lexicalized STSG parsing model with the target lexicalized information.

[6]The LDC category is LDC2003E14.

S2T.Gen.SrcTgt.Lex: This system serves the same function as ***S2T.Gen.Tgt.Lex*** except that it utilizes both the source and target lexicalized information.

S2T.Dis.SrcTgt.Lex: This string-to-tree system integrates the discriminative model with additional bilingual lexicalized features (the boundary words of the constituents in addition to the headword).

Table III shows the results. The linguistically syntax-based string-to-tree system substantially outperforms the hierarchical phrase-based system Joshua and achieves average gains of 2.71 BLEU points[7], demonstrating that the string-to-tree translation model is quite powerful.

Based on the strong string-to-tree model, both the generative and discriminative parsing models improve the translation quality. Specifically, the generative parsing model including the bilingual lexicalized information can significantly outperforms the string-to-tree baseline model by an average of 0.74 BLEU points. In contrast, the improvement obtained by the generative parsing model considering only the target lexicalized information is less promising (0.42 BLEU points). These results show that not only the target-side lexicalized information but also the source-side lexicalized information is useful.

For the discriminative model incorporating additional lexicalized features (last line in Table III), the baseline string-to-tree model is improved by an average of 0.70 BLEU points. However, even using additional lexicalized features (e.g., the boundary words of a constituent), it cannot substantially outperform the generative model. The reason for this result is most likely twofold. On the one hand, in contrast to the generative model, the discriminative model does not consider the structure of the rule (e.g., the order of the nonterminal nodes on the rule's right-hand side). On the other hand, the parameters are determined using a relatively small data set, and the consensus in the machine learning community is that more data are required for the parameter training in discriminative models.

C. Experimental Results on the Large-Scale Data

If a proposed model is useful, then it should also be effective on large-scale data sets. In this section, we report our experimental results on the large data set. The large-scale Chinese-to-English training data set[8] includes 2.09 M sentence pairs from the LDC. The translation rules are extracted using the same settings as the small-scale experiment, except that we obtain the composed rules by combining at most two minimal rules to avoid too many specific rules. The 5-gram language model is trained on the Xinhua portion of the English Gigaword corpus plus the target portion of the bilingual data.

Table IV shows the results. First, we observe that, compared with the small data set, the improvements over MT04 and MT05 are quite different on the larger data set. Using the system ***S2T*** as an example, the improvement over MT04 is 2.67 BLEU points (36.40 vs. 33.73), and the improvement over MT05 is 4.28 BLEU points (34.53 vs. 30.25). We have analyzed the

[7]The average gain is computed by averaging the improvements obtained on MT04 and MT05. For example, $2.71 = ((33.73 - 30.71) + (30.25 - 27.86))/2$.

[8]LDC category numbers are: LDC2000T50, LDC2003E14, LDC2003E07, LDC2004T07, LDC2005T06, LDC2002L27, LDC2005T10 and LDC2005T34.

TABLE IV
EXPERIMENTAL RESULTS FOR DIFFERENT TRANSLATION SYSTEMS FOR LARGE-SCALE DATA SET. $^{+}$ AND $^{++}$ MEAN THAT THE PERFORMANCE IS SIGNIFICANTLY BETTER THAN ***S2T*** WITH $P < 0.05$ AND $P < 0.01$ RESPECTIVELY

Systems	BLEU (%)		
	MT04	MT05	Ave. Gain
Joshua	35.18	32.40	N/A
S2T	36.40	34.53	1.68/N/A
S2T.Gen.Tgt.Lex	36.71	34.98	2.06/0.38
S2T.Gen.SrcTgt.Lex	**36.98**$^{+}$	**35.10**$^{+}$	2.25/0.58
S2T.Dis.SrcTgt.Lex	**37.38**$^{++}$	**35.32**$^{++}$	2.56/0.89

data and find that this result is due primarily to the vocabulary coverage. On the small-scale data set, the vocabulary coverage is 87% and 85% for MT04 and MT05 respectively, whereas on the large-scale data set, the vocabulary coverage is 93% for both MT04 and MT05. The vocabulary coverage improvement is larger for MT05.

Similar to the small-scale experiment, the baseline string-to-tree model ***S2T*** markedly outperforms the hierarchical phrase-based system ***Joshua***, and both the generative and discriminative lexicalized STSG parsing models outperform ***S2T***. The generative parsing model with bilingual lexicalized information, ***S2T.Gen.SrcTgt.Lex***, still produce results superior to those of the generative model with only target-side lexicalized information, ***S2T.Gen.Tgt.Lex***. At the same time it significantly outperforms ***S2T*** with an average gain of 0.58 BLEU points on two test sets. We also observe that the performance improvement for the generative model, ***S2T.Gen.SrcTgt.Lex***, is smaller for the large-scale data set than for the small-scale data set (0.58 vs. 0.74). Our analysis hints that this result may owe to the fact that the generative model utilizes only the headword information. When the data set becomes large, each headword becomes associated with more nonterminals, rendering the prediction from headword to nonterminal more ambiguous. We find that in the small-scale data set, each headword corresponds to 2.13 nonterminals on average, whereas each headword has 2.44 corresponding nonterminals on average in the large-scale data set.

The discriminative parsing model ***S2T.Dis.SrcTgt.Lex*** demonstrates its power in the large-scale experiment, exhibiting the best performance among all of the translation systems. In the large-scale experiment, ***S2T.Dis.SrcTgt.Lex*** outperforms the generative model ***S2T.Gen.SrcTgt.Lex*** on both test sets by an average of 0.31 BLEU points and achieves a significant improvement (0.89 BLEU points on average) over the baseline ***S2T***. These results demonstrate the strength of the discriminative model: the larger training data set yields superior results. For the large-scale data set, we have also conducted an additional experiment for discriminative model. We divided the six features into target-side parts and source-side parts and tested only target features (target headword and boundary words) to determine whether a similar improvement is achieved. We find that the discriminative model with only target features exhibits superior performance compared with the baseline but cannot outperform the discriminative model with all six features. The BLEU scores on MT04 and MT05 are 37.01 and 35.08, respectively.

TABLE V
DISTRIBUTION OF DIFFERENT KINDS OF RULES USED IN DECODING: *LEX*, *N-LEX* AND *MIX* DENOTE RESPECTIVELY PURELY LEXICALIZED RULES, NON-LEXICAL RULES AND TERMINAL/NON-TERMINAL MIXED RULES

Systems	Distribution of Rule Usage					
	MT04			MT05		
	lex	n-lex	mix	lex	n-lex	mix
S2T	.406	.087	.507	.403	.095	.512
S2T.Gen.Tgt.Lex	.443	.044	.513	.441	.032	.527
S2T.Gen.SrcTgt.Lex	.419	.054	.537	.429	.057	.514
S2T.Dis.SrcTgt.Lex	.39	.083	.527	.393	.085	.522

To see the performance when the test genre differs, we split MT08 into two genres: newswire (first 691 sentences) and weblog. ***S2T*** gets BLEU score 29.47 and 21.78 respectively on newswire and weblog, while our discriminative model ***S2T.Dis.SrcTgt.Lex*** obtains 30.43 and 22.26 on newswire and weblog. The improvement on newswire is much larger than that on weblog. We think this is because that our translation rules are extracted mainly on news data.

D. Rule Usage Distribution and Which Rules Need Bilingually Lexicalized STSG the Most?

The translation quality improvement in both the small- and large-scale experiments demonstrates the effectiveness of our proposed bilingually lexicalized STSG. Readers may wonder how the bilingually lexicalized STSG influences the rule usage distribution in decoding and which rules make the most use of the lexicalized information.

We therefore analyze the rule usage in detail in this section. Based on the terminals and nonterminals in a rule, we divide the string-to-tree rules into three categories: (1) *purely lexicalized rules*, such as $\mathrm{NP} \rightarrow (j\check{i}ngf\bar{a}ng,\ \mathrm{the\ police})$; (2) *non-lexical rules*, such as $\mathrm{VP} \rightarrow (\mathrm{x}_0\mathrm{x}_1, \mathrm{VP} : \mathrm{x}_1\mathrm{PP} : \mathrm{x}_0)$; (3) *mixed rules* in which terminals and nonterminals are mixed, such as $\mathrm{VP} \rightarrow (\mathrm{x}_0\ j\check{i}b\grave{i}, \mathrm{were\ killed\ PP} : \mathrm{x}_0)$. We provide statistics regarding the rule usage of the three types of rules for each system (in the large-scale experiments) in Table V.

From the statistics, we can see that for each system, the mixed rules are used most, followed by the purely lexicalized rules. In contrast, the non-lexical rules are not used frequently (they account for only 9.5% of the usage at most). This result is reasonable as the non-lexical rules are usually applied in the construction of the high-level tree structure, which requires fewer rules. The lower-level tree structure is typically constructed using lexicalized and mixed rules when the string-to-tree model generates the target parse tree.

When we compare the bilingually lexicalized string-to-tree models with the baseline, Table V shows that the non-lexical rules are used even less often (from highest 9.5% of the time in the baseline to 3.2% of the time in bilingually lexicalized models), while the mixed rules are applied more frequently. These results imply that given the augmented lexicalized rule probabilities, the bilingually lexicalized models prefer the rules containing lexicalized words when applying rules in decoding.

TABLE VI
TRANSLATION RESULTS WHEN WE DO NOT PERFORM BILINGUALLY-LEXICALIZED PARSING MODELS ON THE PURELY LEXICALIZED RULES (↓ DENOTES THE PERFORMANCE DECLINE COMPARED WITH ORIGINAL RESULTS IN TABLE III)

Systems	BLEU (%)	
	MT04	MT05
S2T.Gen.Tgt.Lex	36.68 (0.03↓)	34.77 (0.22↓)
S2T.Gen.SrcTgt.Lex	36.90 (0.08↓)	35.01 (0.09↓)
S2T.Dis.SrcTgt.Lex	37.29 (0.09↓)	35.16 (0.16↓)

TABLE VII
TRANSLATION RESULTS WHEN WE ONLY PERFORM BILINGUALLY-LEXICALIZED PARSING MODELS ON THE NON-LEXICAL RULES (↑ DENOTES THE PERFORMANCE IMPROVEMENT COMPARED WITH THE BASELINE SYSTEM ***S2T***)

Systems	BLEU (%)	
	MT04	MT05
S2T	36.40	34.53
S2T.Gen.Tgt.Lex	36.49 (0.09↑)	34.65 (0.12↑)
S2T.Gen.SrcTgt.Lex	36.72 (0.32↑)	34.79 (0.26↑)
S2T.Dis.SrcTgt.Lex	36.83 (0.43↑)	34.90 (0.37↑)

Another question arises regarding the type of rules that make the most use of the bilingually lexicalized model. Intuitively, the non-lexical and mixed rules both contain nonterminals that are generalized and require lexicalization to distinguish good rules from poor ones, while the purely lexicalized rules are fully lexicalized from the outset. In principle, the purely lexicalized rules therefore do not require any bilingually lexicalized model. To investigate this question, we repeat the large-scale experiments with the bilingually lexicalized models used only for the non-lexical and mixed rules. Table VI shows the translation results.

It is interesting that the results in Table VI are in line with our intuition: the decrease in the performance is small for every system when we do not apply bilingually lexicalized parsing models to the purely lexicalized rules (the largest drop of 0.22 BLEU points is not significant). We can conclude that the non-lexical rules and mixed rules need the bilingually lexicalized models most.

We conduct an additional experiment to test the performance when the bilingually lexicalized parsing models are applied only to the non-lexical rules. Table VII shows the results on the large-scale data. The generative lexicalized parsing models (***S2T.Gen.Tgt.Lex*** and ***S2T.Gen.SrcTgt.Lex***) improve the translation quality only slightly compared with the baseline ***S2T*** (the largest improvement is 0.32 BLEU points). The discriminative lexicalized parsing model achieves larger gains compared with the baseline, and the largest improvement is 0.43 BLEU points. Note that the improvement is somewhat promising because the non-lexical rules comprise only a small fraction of the rules used in decoding (not more than 8.5%, as shown in Table V). The experimental results demonstrate that the bilingually lexicalized parsing models are effective in rendering the non-lexical rules less ambiguous during the rule selection in the decoding process.

TABLE VIII
PERCENTAGE OF COMPOSED RULES USED DURING DECODING. THE NUMBER IN PARENTHESES DENOTES THE PROPORTION OF NON-LEXICAL AND MIXED RULES AMONG THE APPLIED COMPOSED RULES

Systems	Usage of composed rules	
	MT04	MT05
S2T	22.9% (68.7%)	21.51% (65.8%)
S2T.Gen.Tgt.Lex	16.8% (72.3%)	15.68% (71.5%)
S2T.Gen.SrcTgt.Lex	16.1% (75.6%)	15.40% (74.1%)
S2T.Dis.SrcTgt.Lex	22.0% (70.2%)	20.87% (72.9%)

When we compare the numbers in Table VII with those in Table VI, we find a modest gap; the largest gain is 0.46 BLEU points (36.83 vs. 37.29). This result indicates that the bilingually lexicalized parsing models also contribute substantially to the mixed rules.

In conclusion, both the non-lexical and mixed rules make substantial use of the bilingually lexicalized parsing models.

E. Relationship With the Composed Rules

The STSG rules in the string-to-tree models can also be divided into *minimal rules* and *composed rules*. It may be argued that composed rules already encode substantially more lexicalization compared with minimal rules, and the use of the composed rules therefore weakens the need for bilingually lexicalized models in string-to-tree translation.

We argue that the composed rules are not a competitor, but rather a complement to the bilingually lexicalized models. This relationship is reflected in the following two points:

On the one hand, they play different roles during decoding. The composed rules are designed to improve the rule coverage and encode more contexts [8], while our bilingually lexicalized models aim to distinguish effective rules from poor ones by dynamically associating the inside nonterminals with source- and target-side lexicalized information during decoding.

On the other hand, as long as the composed rules contain nonterminals on their right-hand sides, bilingually lexicalized models can help to distinguish them from other rules in principle. We therefore examine the decoding process to determine how many composed rules are used and how many of the utilized composed rules contain nonterminals. In the large-scale experiment, the composed rules comprise approximately 20% of all the used rules utilized, and most of the applied composed rules are non-lexical and mixed ones that contain nonterminals. Table VIII shows the statistics. Based on the analysis in the previous section, this result indicates that the bilingually lexicalized models can be useful for the composed rules. That is, the composed rules need the bilingually lexicalized models.

F. Decoding Efficiency of Bilingually Lexicalized STSG Models

The incorporation of lexicalized STSG is disadvantageous in terms of the decoding speed because headwords and boundary words are dynamically acquired, and multiple probabilities must be calculated.

Table IX lists the average decoding time for each system on the two test sets in the large-scale experiments. All of the experiments are conducted using the same hardware: 2.4 GHz ×

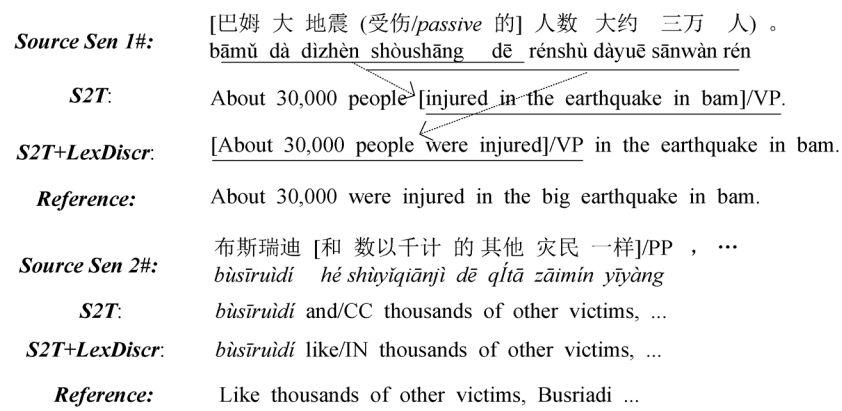

Fig. 5. Two real translation examples, $bùsīruìdí$ in the second example is an unknown word.

TABLE IX
AVERAGE DECODING TIME IN WORDS PER SECOND

Systems	Decoding Time (words/sec)
	MT04-05
S2T	1.02
S2T.Gen.Tgt.Lex	0.58
S2T.Gen.SrcTgt.Lex	0.45
S2T.Dis.SrcTgt.Lex	0.72

$2CPU \times 64$ GB *memory*. The string-to-tree translation is relatively slow because the decoding search space is large, and the system searches all of the possible target tree structures. The incorporation of the lexicalized STSG models further reduces the decoding speed. The generative model with bilingually lexicalized information, ***S2T.Gen.SrcTgt.Lex***, is the slowest; this model decreases the decoding speed by 55.9% compared with the baseline. This result most likely owes to the fact that the model must calculate more backoff probabilities when computing the lexicalized rule probability. In contrast, the discriminative model, ***S2T.Dis.SrcTgt.Lex***, exhibits slightly superior performance and slows down the baseline by only 29.4%. Based on the translation results and decoding speed, the discriminative model is more efficient and more effective compared with the generative models.

G. Translation Examples

To facilitate a better intuition to the ability of our proposed bilingually lexicalized parsing models against the baseline, we present two actual translation outputs produced by the baseline string-to-tree system ***S2T*** and the discriminative bilingually lexicalized system ***S2T.Dis.SrcTgt.Lex*** in the large-scale experiments in Fig. 5. The discriminative model is chosen because it exhibits the best performance in all of the bilingually lexicalized models.

In the first example, *injured* and *were injured* are two possible translations of the single Chinese word $shòushāng$. However, only the translation *were injured* can produce a well-formed target tree structure because the Chinese word in this sentence indicates a passive meaning. When we investigate the optimal derivation path for these two systems, we find that the baseline model transformed the first three Chinese words, $bāmǔ\ dà\ dìzhèn$, into an English prepositional phrase (PP). The next two words, $shòushāng\ \ dē$, are transformed into a verb phrase (VP). The rule $\mathrm{VP} \rightarrow (\mathrm{x_0 x_1}, \mathrm{VP} : \mathrm{x_1 PP} : \mathrm{x_0})$ is then used to translate the first five words into a larger verb phrase, *injured in the earthquake in bam*; the passive word *were* is neglected. The baseline model produced this error because it constructed the high-level tree structure without considering the source- and target-side lexicalized information. In contrast, using the bilingually lexicalized information in a discriminative fashion, the system ***S2T.Dis.SrcTgt.Lex*** properly transformed the last six words into the English verb phrase *about 30,000 people were injured* and obtained the correct translation.

In the second example, the Chinese word ***h\acute{e}*** is in fact a preposition, and the Chinese string $hé\ shùyǐqiānjì\ dē\ qítā\ zāimín\ yīyàng$ is a prepositional phrase. Without the source-side syntactic information, it is reasonable to translate the Chinese word $hé$ into the English conjunction *and*. However, when the following context is taken into account, we have sufficient information to translate the Chinese word $hé$ into an English preposition. Because the boundary lexicalization (e.g., the words $hé$ and $yīyàng$) is not considered when the baseline model constructs the high-level tree structures, the baseline system translated the Chinese string $hé \ldots yīyàng$ as a part of a noun phrase. Fortunately, our proposed system ***S2T.Dis.SrcTgt.Lex*** yielded a correct translation as it considers the headword and boundary words on both the source- and target-sides throughout the tree construction process.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel bilingually lexicalized STSG parsing model to achieve improved translation rule selection in STSG-based string-to-tree decoding. We first proposed a generative model for incorporating bilingual lexicalized information (e.g., headwords). We then proposed a simple discriminative model to incorporate additional bilingual lexicalized features. Our experimental results demonstrate that our approach can significantly improve the translation quality on both small-scale and large-scale data sets. The results also imply that both the target- and source-side lexicalized information is essential in the improvement of the translation quality. Furthermore, the bilingually lexicalized parsing models positively influence the rule distribution in decoding, favoring rules with lexical words as constraints, and we find that the rules containing

nonterminals on their right-hand side need the bilingually lexicalized STSG parsing model the most.

In future work, we intend to study the discriminative model in greater detail in order to explore its potential. For instance, we will explore additional informative features and investigate whether non-lexicalized parsing models used in the monolingual parsing community can inform our modeling strategies. In addition, we plan to investigate the use of bilingually lexicalized STSG in tree-to-tree translation models, which also lack lexicalization information in the construction of the high-level target-side tree structures.

REFERENCES

[1] E. Charniak, K. Knight, and K. Yamada, "Syntax-based language models for statistical machine translation," in *Proc. MT Summit IX*, 2003, pp. 40–46.

[2] D. Chiang, "Hierarchical phrase-based translation," *Comput. Linguist.*, vol. 33, no. 2, pp. 201–228, 2007.

[3] D. Chiang, K. Knight, and W. Wang, "11,001 new features for statistical machine translation," in *Proc. NAACL '09*, 2009, pp. 218–226.

[4] D. Chiang, "Learning to translate with source and target syntax," in *Proc. ACL '10*, 2010, pp. 1443–1452.

[5] D. Chiang and K. Knight, "An introduction to synchronous grammars," in *Tutorial ACL '06*, 2006.

[6] M. Collins, "Head-driven statistical models for natural language parsing," *Comput. Linguist.*, vol. 29, no. 4, pp. 589–637, 2003.

[7] J. Eisner, "Learning non-isomorphic tree mappings for machine translation," in *Proc. ACL '03*, 2003, pp. 205–208.

[8] M. Galley, J. Graehl, K. Knight, D. Marcu, S. Deneefe, W. Wang, and I. Thayer, "Scalable inference and training of context-rich syntactic translation models," in *Proc. COLING-ACL '06*, 2006, pp. 961–968.

[9] M. Galley, M. Hopkins, K. Knight, and D. Marcu, "What's in a translation rule," in *Proc. NAACL '04*, 2004, pp. 273–280.

[10] L. Huang, K. Knight, and J. A. Joshi, "A syntax-directed translator with extended domain of locality," in *Proc. AMTA '06*, 2006, pp. 1–8.

[11] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proc. EMNLP '04*, 2004, pp. 388–395.

[12] Z. Li, C. Callison-Burch, C. Dyer, J. Ganitkevitch, S. Khudanpur, L. Schwartz, W. Thornton, J. Weese, and O. Zaidan, "Joshua: An open source toolkit for parsing-based machine translation," in *Proc. ACL '09*, 2009, pp. 135–139.

[13] Y. Liu, Y. Lv, and Q. Liu, "Improving tree-to-tree translation with packed forests," in *Proc. ACL '09*, 2009, pp. 558–566.

[14] Y. Liu and Q. Liu, "Joint parsing and translation," in *Proc. COLING '10*, 2010, pp. 707–715.

[15] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proc. COLING-ACL '06*, 2006, pp. 609–616.

[16] D. Marcu, W. Wang, A. Echihabi, and K. Knight, "SPMT: Statistical machine translation with syntactified target language phrases," in *Proc. EMNLP '06*, 2006, pp. 44–52.

[17] H. Mi, L. Huang, and Q. Liu, "Forest-based translation," in *Proc. ACL-08: HLT*, 2008, pp. 192–199.

[18] F. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, "A smorgasbord of features for statistical machine translation," in *Proc. NAACL '04*, 2004, pp. 161–168.

[19] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," in *Proc. COLING-ACL '06*, 2006, pp. 433–440.

[20] M. Post and D. Gildea, "Language modeling with tree substitution grammars," in *Proc. NISP Workshop Grammar Induct., Represent. of Lang., Lang. Learn.*, 2009, pp. 1–8.

[21] C. Quirk, A. Menezes, and C. Cherry, "Dependency treelet translation: Syntactically informed phrasal SMT," in *Proc. ACL '05*, 2005, pp. 271–279.

[22] L. Shen, J. Xu, and R. Weschedel, "A new string to dependency machine translation algorithm with a target dependency language model," in *Proc. ACL-08:HLT*, 2008, pp. 577–585.

[23] X. Wu, W. Matsuzaki, and J. Tsujii, "Effective use of function words for rule generalization in forest-based translation," in *Proc. ACL '10*, 2010, pp. 22–31.

[24] T. Xiao, J. Zhu, and M. Zhu, "Language modeling for syntax-based machine translation using tree substitution grammars: A case study on Chinese-English," in *Proc. ACM Trans. Asian Lang. Inf. Process.*, 2011.

[25] J. Xie, H. Mi, and Q. Liu, "A novel dependency-to-string model for statistical machine translation," in *Proc. EMNLP '11*, 2011, pp. 216–226.

[26] D. Xiong, Q. Liu, and S. Lin, "Maximum entropy based phrase reordering model for statistical machine translation," in *Proc. ACL-COLING '06*, 2006, pp. 521–528.

[27] D. Xiong, Q. Liu, and S. Lin, "A dependency treelet string correspondence model for statistical machine translation," in *Proc. WMT '07*, 2007, pp. 40–47.

[28] K. Yamada and K. Knight, "A syntax-based statistical translation model," in *Proc. ACL '01*, 2001, pp. 523–530.

[29] H. Zhang, L. Fang, P. Xu, and X. Wu, "Binarized forest to string translation," in *Proc. ACL '11*, 2011, pp. 835–845.

[30] H. Zhang, L. Huang, D. Gildea, and K. Knight, "Synchronous binarization for machine translation," in *Proc. HLT-NAACL '06*, 2006, pp. 256–263.

[31] H. Zhang and D. Gildea, "Stochastic lexicalized inversion transduction grammar for alignment," in *Proc. ACL '05*, 2005, pp. 475–482.

[32] H. Zhang and D. Gildea, "Inducing word alignments with bilexical synchronous trees," in *Proc. COLING/ACL '06*, 2006, pp. 953–960.

[33] J. Zhang, F. Zhai, and C. Zong, "Augmenting string-to-tree translation models with fuzzy use of source-side syntax," in *Proc. EMNLP '11*, 2011, pp. 204–215.

[34] L. Zhang, 2004, Maximum Entropy Modeling Toolkit for Python and C++, [Online]. Available: http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

[35] M. Zhang, H. Jiang, A. Aw, H. Li, C. Tan, and S. Li, "A tree sequence alignment-based tree-to-tree translation model," in *Proc. ACL-08: HLT*, 2008, pp. 559–567.

[36] A. Zollmann and S. Vogel, "A word-class approach to labeling PSCFG rules for machine translation," in *Proc. ACL '11*, 2011, pp. 1–11.

[37] S. DeNeefe and K. Knight, "Synchronous tree adjoining machine translation," in *Proc. EMNLP '09*, 2009, pp. 727–736.

[38] F. Zhai, J. Zhang, Y. Zhou, and C. Zong, "Tree-based translation without using parse trees," in *Proc. COLING '12*, 2012, pp. 3037–3054.

[39] F. Zhai, J. Zhang, Y. Zhou, and C. Zong, "Simple but effective approaches to improving tree-to-tree model," in *Proc. MT Summit XIII*, 2012, pp. 261–268.

Jiajun Zhang received the B.Sc. degree from Jilin University, Changchun, China, in 2006 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2011, both in computer science.

Since 2011, he has been with the National Laboratory of Pattern Recognition, which is a part of the Institute of Automation, Chinese Academy of Sciences, Beijing, China, as an Assistant Professor. His research interests include statistical machine translation, natural language processing.

Feifei Zhai graduated in 2009 from Beijing Jiaotong University, China and then became a Ph.D. candidate of the Institute of Automation, Chinese Academy of Sciences (CASIA). His current research direction is machine translation.

Chengqing Zong is a professor in natural language technology at the National Laboratory of Pattern Recognition and the deputy director of the National Laboratory of Pattern Recognition, which is part of the Chinese Academy of Sciences' Institute of Automation. His research interests include machine translation, text classification, and the fundamental research on Chinese language processing. Zong received his PhD from the Chinese Academy of Sciences' Institute of Computing Technology in March 1998. He is a member of the International Committee on Computational Linguistics, and Chair-Elect of SIGHAN, ACL.