Syntax-Based Chinese-Vietnamese Tree-to-Tree Statistical Machine Translation with Bilingual Features

SHENGXIANG GAO, JIHAO HUANG, MINGYA XUE, ZHENGTAO YU, ZHUO WANG, and YANG ZHANG, School of Information Engineering and Automation, Kunming University of Science and Technology, China

Because of the scarcity of bilingual corpora, current Chinese–Vietnamese machine translation is far from satisfactory. Considering the differences between Chinese and Vietnamese, we investigate whether linguistic differences can be used to supervise machine translation and propose a method of syntax-based Chinese–Vietnamese tree-to-tree statistical machine translation with bilingual features. Analyzing the syntax differences between Chinese and Vietnamese, we define some linguistic difference-based rules, such as attributive position, time adverbial position, and locative adverbial position, and create rewards for similar rules. These rewards are integrated into the extraction of tree-to-tree translation rules, and we optimize the pruning of the search space during the decoding phase. The experiments on Chinese–Vietnamese bilingual sentence translation show that the proposed method performs better than several compared methods. Further, the results show that syntactic difference features, with search pruning, can improve the accuracy of machine translation without degrading the efficiency.

CCS Concepts: • **Computing methodologies** → Machine translation;

Additional Key Words and Phrases: Statistical machine translation, tree-to-tree, Chinese-Vietnamese, linguistic features, pruning optimization

ACM Reference format:
Shengxiang Gao, Jihao Huang, Mingya Xue, Zhengtao Yu, Zhuo Wang, and Yang Zhang. 2019. Syntax-Based Chinese-Vietnamese Tree-to-Tree Statistical Machine Translation with Bilingual Features. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 18, 4, Article 36 (May 2019), 20 pages.
https://doi.org/10.1145/3314938

1 INTRODUCTION

Machine translation is the most effective way to promote cross-language communication. In recent years, data-driven machine-translation methods, such as statistical machine translation

This work was supported by National Natural Science Foundation of China (Grant Nos. 61732005, 61672271, 61761026, 61762056, 61472168), National Key Research and Development Plan (Grant Nos. 2018YFC0830105, 2018YFC0830100), Science and Technology Leading Talents in Yunnan, and Yunnan High and New Technology Industry Project (Grant No. 201606), Natural Science Foundation of Yunnan Province (Grant No. 2018FB104), and Talent Fund for Kunming University of Science and Technology (Grant No. KKSY201703005).
Authors' addresses: S. Gao, J. Huang, M. Xue, Z. Yu (corresponding author), Z. Wang, and Y. Zhang, School of Information Engineering and Automation, Kunming University of Science and Technology, No. 727 South Jingming Rd., Chenggong District, 650500, Kunming, China; emails: gaoshengxiang.yn@foxmail.com, {huangjihao001, kdxuemingya, wangzhuo}@163.com; ztyu@hotmail.com, yoummg@qq.com.

2375-4699/2019/05-ART36 $15.00
https://doi.org/10.1145/3314938

and neural machine translation, have made good progress. For neural machine translation, Kalchbrenner et al. [1] proposed a translation model based on recurrent neural networks. It unified learning and modeling for encoding and decoding. Sutskever et al. [2] and Bahndanau et al. [3] introduced an attention mechanism to improve the performance of RNN model greatly. Much research on modeling and knowledge fusion in neural machine translation has also made great progress [4–6]. Based on the scale of tens of millions of bilingual corpora, Google has launched a variety of world mainstream language neural machine-translation systems including Chinese-English translation, which basically achieves or closes the level of human translation in some translation tasks [7]. Neural machine translation has a significant effect under large-scale bilingual sentence pairs. But for resource-poor languages, such as Chinese-Vietnamese, Chinese-Thai, Chinese-Burmese, Chinese-Cambodian, Chinese-Laotian, and so on, due to the scarcity of bilingual sentence pairs, neural machine translation has not shown to be good or effective.

Syntax-based statistical machine translation is in the mainstream of statistical machine-translation research in recent years. The main idea is to introduce syntactic constraints on the basis of traditional phrase-based translation model to improve the effect of statistical machine translation. Chiang et al. [8] proposed a hierarchical phrase-based statistical machine-translation model that uses synchronous context-free grammar. Galley et al. [9] proposed a context-rich syntax-based translation model. Zhang et al. [10] extended a string-to-tree model using fuzzy methods. Liu et al. [11] proposed a tree-to-string alignment template, taking source-side syntax into consideration. In addition, References [12–16] proposed some syntax-based machine-translation methods based on tree-to-string, tree-to-tree, and sub-tree alignment models, respectively. These models mainly explore the use of different syntactic information and have achieved good translation results.

Chinese-Vietnamese machine translation is a typical resource-scarce language machine translation. At present, there is relatively little research work, and many studies mainly focus on phrase-based statistical machine translation. Tran et al. [17] mainly explored a statistical machine-translation method based on character-level and word-level of Chinese and Vietnamese. They also carried out in-depth research on unknown words of named entities and organization names and achieved good results. References [18–20] fused Vietnamese postposition characteristics to a decoding process of a phrase-based machine-translation model. As a result, it has enhanced the performance of Chinese-Vietnamese machine translation. In addition, Ha [21] proposed a pivot-based Chinese-Vietnamese machine-translation method. It used Chinese as the pivot to translate Vietnamese syllables into Chinese characters one-by-one. After, the Chinese characters, under the application of Chinese grammar, were combined as grammatical sentences. At present, corporations such as Google, Baidu, Microsoft, and Sogou have developed Chinese-Vietnamese online translation systems.[1] But, in terms of accuracy, they are still far from practical.

Due to the difficult construction of a Chinese and Vietnamese bilingual corpus and the relatively high cost, Chinese-Vietnamese machine translation still faces the problem of small-scale bilingual corpus. Compared with language structure, Chinese and Vietnamese have certain differences in language grammar, such as, in Vietnamese, attributive postposition and adverbial postposition; that modifier to a central word has a certain order of modification and so on. This syntactic knowledge has an important impact on bilingual translation, or should be said to translation modeling. It is a worthwhile discussion to integrate language-difference features into model training and the decoding process to improve translating accuracy and performance. Therefore, on the basis of an in-depth analysis to bilingual language differences, this article proposes a syntax-based

[1]Google online translation: http://translate.google.com.
Baidu translator: http://fanyi.baidu.com/translate.
Sogou translation: http://fanyi.sogou.com.

Table 1. Vietnamese-Chinese Word Order Differences

Chinese	Vietnamese
她是我见过的最美丽的女孩 (She is the most beautiful girl I've ever seen.)	Cô là "她是 (She is)" cô gái "女孩 (girl)" đẹp "美丽的 (beautiful)" nhất "最 (most)" mà tôi từng thấy "我见过的 (I've ever seen)"
孩子们很喜欢那种上面有字的白色的心状糖果 (The children are very fond of the white heart-shaped candy with the words on it.)	Trẻ con "孩子们 (The children)" rất "很 (very)" thích "喜欢 (fond of)" loại "那种 (the)" kẹo "心状糖果 (heart-shaped candy)" màu trắng "白色 (white)" ben trên có chữ "上面有字 (words on it)"

Chinese-Vietnamese tree-to-tree statistical machine-translation method. It makes full use of both source-side syntax and target-side syntax, fuses bilingual features, and constructs a syntax-based translation model to improve the performance of Chinese-Vietnamese machine translation.

The main contributions of this article are as follows:

(1) Linguistic differences between Chinese and Vietnamese are analyzed and formalized to fit a log-linear translation model.
(2) The minimal rule extraction is introduced, and rewards or penalties based on linguistic differences are used to improve the accuracy of rule extraction.
(3) The decoding performance of the model is improved by pruning based on linguistic features.

The proposed method is compared with several baseline versions. The results show that syntactic difference features along with search pruning can improve the accuracy of Chinese-Vietnamese machine translation without degrading the efficiency of the method.

2 CHINESE-VIETNAMESE BILINGUAL DIFFERENCES AND FORMALIZATION

Chinese is a Sino-Tibetan language family, and Vietnamese is an Austroasiatic language. Chinese and Vietnamese have similarities and differences in grammar. They can guide translation well and require detailed analysis and formal definition.

The greatest similarity between the two languages is that their main sentence components are in the same order. That is, they both have subject–verb–object structure. For example, a Chinese sentence "她是一个女孩(she is a girl)" translated to Vietnamese is "Cô(she) là (is) cô gái (girl)."

The most significant difference between the two languages is that the order of the modifier word and central word is completely opposite [22]. For example, the Chinese phrase, "美丽的女孩 (beautiful girl)" is "cô gái (girl) đẹp (beautiful)" in Vietnamese. In Chinese, the order of descriptive multi-tiered attributive modifiers is as follows: (i) predicate phrases, (ii) verbs (phrases)/prepositional phrases, (iii) adjective phrases and other descriptive phrases, and (iv) adjective and descriptive nouns without "的." In Vietnamese, the order of descriptive multi-tiered attributive modifiers is the reverse. That is, the order of descriptive attributives in Chinese is (i), (ii), (iii), and (iv), while in Vietnamese it is (iv), (iii), (ii), and (i). Table 1 shows two examples of word orders in Vietnamese and in Chinese.

Vietnamese grammatical features can be summarized as follows:

(1) An adverb modifying a verb is located behind the verb.
(2) An adverb modifying an adjective is located behind the adjective.
(3) An adjective modifying a noun or nominal phrase is located behind the noun or nominal phrase.

(4) When multiple adjectives modify a noun or noun phrase, they are placed behind the modified noun, and their order is the reverse order of the corresponding Chinese sequences.

Therefore, two general translation principles can be obtained: (i) In Vietnamese, all the modifiers are located behind the modified word. And the modifier and the modified word are sequential. (ii) In Chinese, all the modifiers are located before the modified word. And the modified word and the modifier are sequential. For the grammatical differences between Chinese and Vietnamese, the following three feature rewards are proposed:

(1) Attributive postposition reward AD(d). In contrast to Chinese, Vietnamese attributive position is usually behind the main word, as follows rule:
NP(ADJP NP)→ NP(NP ADJP).
(2) Time adverbial postposition reward TD(d). Time adverbial in Vietnamese occurs, at the opposite order to Chinese sequences, behind the main word, as follows rule:
NP(NT VP)→ NP(VP NT).
(3) Locative adverbial postposition reward LD(d). In contrast to Chinese, the locative adverbial in Vietnamese is usually located behind the predicate verb as follows rule:
NP(ADVP NP)→NP(NP ADVP).

Three corresponding template rules are put into the template corpus. Here, NP, ADJP, NT, VP, and ADVP stand for noun phrase, adjective phrase, noun time, verb phrase, and adverbial phrase, respectively.

In training parameter phase, the templates that conform to the template corpus are rewarded and the other rules remain unrewarded. This not only ensures the integrity of the rules, but also improves their accuracy.

3 THE PROPOSED TRANSLATION MODEL

To make full use of syntactic information and linguistic differences between Chinese and Vietnamese, this article, on the basis of tree-to-tree statistical machine-translation model, explores the integration of Chinese-Vietnamese linguistic differences with a syntax-based translation model. There are many advantages to using a syntax-based model when only a small amount of data is available. For example, consider the Chinese phrase “老挝国会主席” and assume that the word “老挝” does not appear in the training corpus. In this case, if we use a word- or phrase-based model, the error that occurs when decoding “老挝” can affect the following decoding of “国会” and “主席”. In contrast, when using a syntax-based model, the decoding is totally different. Decoding from a derivation is likely to generate the NP node properly, and as a result, the error that occurs when decoding “老挝” would not affect the remainder of the phrase. In addition, using syntax in a translation model can enable a complicated reordering problem to be solved.

The work, at first, preprocesses during tree-to-tree translation rule extraction to remove those rules that do not conform to the language differences. In the model training phase, it rewards those features that satisfy the translation rules and increases their weights. Strong constraints on the syntax tree lead to too few rules, which reduce its ability to handle more varied linguistic information. Hence, we use the strategy of combining and generalizing rules to expand the rule base; that is, to extract the smallest rules, to formalize the conditions of combining them, to combine them, and then utilize the source language phrase information to do fuzzy matching, and finally to expand the rule base. In the process of decoding, we optimize the pruning of candidate translations that do not conform to the language differences and integrate the language differences into the model to improve the Chinese-Vietnamese machine-translation efficiency.

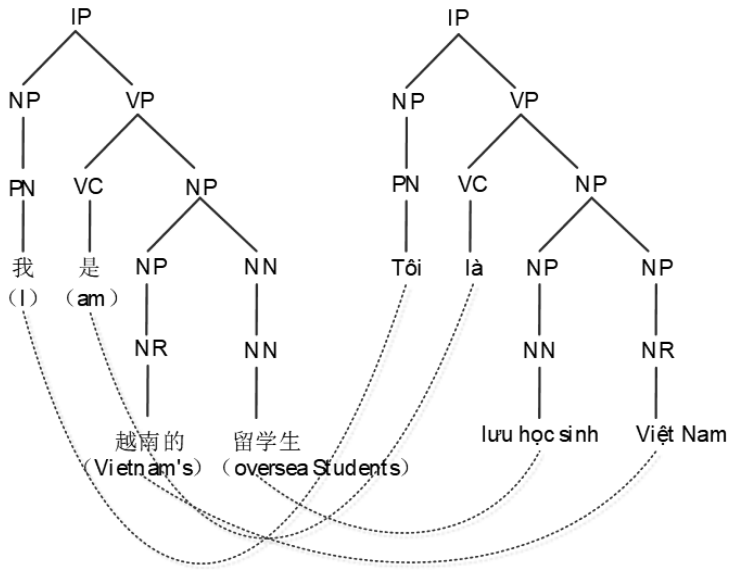

Fig. 1. Chinese-Vietnamese phrase syntax tree alignment information.

Table 2. Chinese-Vietnamese Syntax Rules Extracted from the Example in Figure 1

Source side rule → Target side rule
PN(I) → PN(Tôi)
VP(VC(am)NP) → VP(VC(là)NP)
NP(NR(Vietnamese)NN(oversea student) → NP(NN(lưu học sinh)NR(Việt Nam))

3.1 Alignment and Rule Extraction

3.1.1 Minimal Rule Extraction and Optimization. To extract the translation rules, the Chinese and Vietnamese languages were syntactically analyzed using 100,000 aligned words and 50,000 aligned sentences from a Chinese and Vietnamese bilingual corpus, which was prepared by researchers in our laboratory. Then, according to the concepts presented in Reference [9], the result of word alignment was used to obtain the correspondence between the source- and target-language tree nodes. In addition, the syntax sub trees are extracted by finding the boundary nodes. The cut boundary position (node) can clearly distinguish between different sub-tree fragments so the alignment between the source or target language fragments can be determined. A one-to-one correspondence between the nonterminal boundary nodes of the smallest sub-tree fragment of any root node on both sides is needed to construct a rule.

Figure 1 shows an example of a Chinese-Vietnamese sentence tree alignment. The Chinese sentence is "我是越南留学生" (I am a Vietnamese student), and the Vietnamese version is "tôi 我(I) là 是(am) lưu học sinh 留学生 (overseas student) Việt Nam 越南(Vietnamese)." Using the method above, the rules can be extracted and then synchronous tree substitution grammar can be used to deduce the minimum rules. The obtained rules are shown in Table 2.

We define the corresponding features on each existing rule to calculate the translation probabilities.

However, the strong constraints of the syntax tree will lead to some extracted rules not conforming to the Chinese-Vietnamese grammatical differences. These rules will introduce many errors

into the decoding process. Hence, we should remove the rules that do not conform to the language characteristics before the rule extraction completion.

To use the above inductive three template rules, attributive postposition, time adverbial postposition, and locative adverbial postposition, we define an evaluation function ∂ to evaluate the extracted rules.

$$\partial = \begin{cases} NP(ADJP\ NP) \rightarrow NP(NP\ ADJP) \\ NP\ (NT\ VP) \rightarrow NP(VP\ NT) \\ NP(ADVP\ NP) \rightarrow NP(NP\ ADVP) \\ delete \end{cases}. \tag{1}$$

Rule evaluation is divided into two steps:

(1) Identify a rule; if it conforms to the first three expressions in function ∂, then to do continuous matching; otherwise, the rule is ignored.
(2) The continuous matching rule should be deleted once it does not conform to the definition.

3.1.2 Rule Combination. When extracting syntax tree rules, the principle of rule combination synchronously combines the extracted minimum number of rules. A minimum rule is defined as one that is extracted only from the boundary nodes; that is, it is a fragment of the sub tree. However, each tree often contains a large amount of syntactic information, so we should use a combination of rules to expand the rule base. The algorithm for combined minimum extraction rules uses dynamic programming. Because of limited computing resources, we present four constraints for a rule combination:

(1) The depth of a combination rule cannot exceed the height of tree h.
(2) The terminal number of leaf nodes cannot exceed c.
(3) The number of trees in a rule is not greater than d.
(4) A combination rule is a collection of up to two or three initial rules.

Here, h is the original syntax tree. These constraints can effectively improve the efficiency of the algorithm. To a certain extent, the size of the rule base is enlarged, and the model can be easily implemented by adjusting the values of parameters c and d.

For example, in Table 2, we have rule 2: VP (VC "是 (am)" NP) → VP (VC (là) NP) and rule 3: NP (NR "越南(Vietnamese)" NN "留学生 (Oversea Student)" → NP (NN (lưu hṇc sinh) NR (Việt Nam)), on syntax tree VP (VC NP). We can combine these rules to produce the following combination rule 4: VP(VC "是(am)" NP((NR "越南(Vietnamese)" NN "留学生 (Oversea Student)")) → VP (VC (là) NP (NN (lưu hṇc sinh) NR (Việt Nam))).

3.1.3 Rule Generalization. On a limited Chinese-Vietnamese corpus, the syntax rules extracted from the tree-based translation model are also limited. To expand the coverage, we propose two simple fuzzy-matching-based methods to transform the source language phrases into tree-to-tree translation rules and enrich the extracted rule base.

References [11] and [23] use bilingual phrases to improve the performance of tree-to-string models and forest-to-string models. However, it is difficult to integrate bilingual phrases into a tree translation model to solve the problem of poor rule coverage. Therefore, we use the syntax structure of the source and target languages to facilitate the decoding process and translate the source language phrases into tree-to-tree translation rules, which are more easily integrated into our tree-to-tree model.

In traditional tree-based decoding, the source-end rules exactly match the source tree. Therefore, if we want to use a source language phrase, theoretically, the corresponding grammatical structure, such as a tree-based model, must be used to express it. However, experience shows that

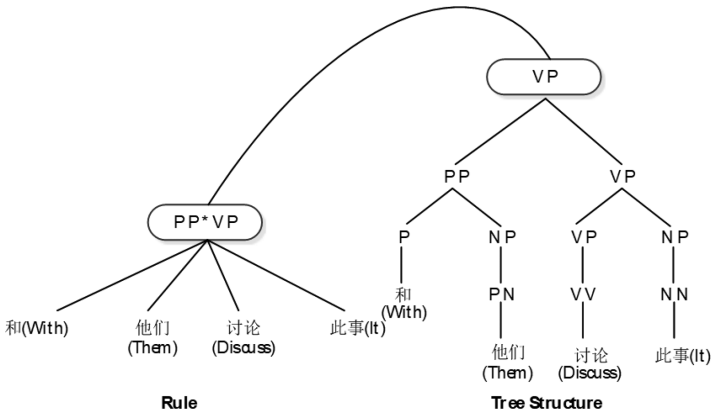

Fig. 2. Rule generalization.

accurate matching can harm the quality of the translation. Therefore, instead of using grammatical structure, an appropriate grammar category will be better, which has been proved that this is necessary and effective for translation [24]. When decoding with these source phrases, the internal structure of the translated sub tree is ignored as long as the leaf nodes and root node match the rules of the sub tree, as shown in Figure 2.

We use Syntax Augmented Machine Translation via chart parsing (SAMT) proposed by Reference [25], where each phrase can be associated with the corresponding syntax category. For example, as shown in Figure 2, "discuss it with them" is not a syntax sub tree, but we can define the rule PP*VP using the annotation method in Reference [24].

Normally, if we perform an exact match, we do not enumerate this rule because rule C does not match the tree structure. We maximize the source language phrase information using the fuzzy matching method, which has been successfully applied in translation models based on hierarchical phrases [26] and the string-to-tree model [24]. Using the fuzzy matching method, we express each syntax rule of SAMT with a vector F, which captures the underlying syntactic information. Drawing on the methods proposed by Reference [26], the similar degree between two syntax rules can be calculated by using dot product as follows:

$$\bar{F}(c) \cdot \bar{F}(c') = \sum_{1 \le i \le n} f_i(c) f_i(c'). \tag{2}$$

Where $\overline{\mathrm{F}}(\mathrm{c})$ and $\overline{\mathrm{F}}(\mathrm{c}')$ are normalized feature vectors, representing the tag sequences of c and c', respectively. c is a syntax rule and c' is another syntax rule. The advantage of using real-valued feature vectors is that the degree of similarity between two tag sequences in the space of latent syntax categories can be simply computed as the dot product of their feature vectors. This produces a similarity score range from 0 (completely different syntax) to 1 (completely identical syntax).

Using the SAMT syntax, a corresponding real vector represents the original source language phrase. In the decoding process, we consider all possible source language phrases to calculate the similarity between the phrase category and the syntax structure for a head node. Then, the similarity score is incorporated into the model as a function (called a similarity score function).

3.2 Feature Set and Parameter Training

3.2.1 Definition of a Feature Set. We also need to define the features of each rule derivation and use them to calculate translation probabilities. Following References [8] and [27], we consider the following features of the overall translation process:

(h1) Phrase translation probability. A translation probability of leaf-node sequences that correspond to source language sub tree and target language sub tree, respectively.
(h2) Lexical weight. That is the intensity between word corresponding relationships of leaf nodes. Given a rule and the word alignment between the corresponding source and target words, we can define the lexical weight from the source language to the target language as follows:

$$\Pr_{lex}(\overline{t_r}|\overline{s_r}, a) = \prod_{i=1}^{m} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall(i,j)\in a} w(t_i|s_j). \tag{3}$$

(h3) Rule probability based on the root node syntax tag, denoted as $\Pr(r|\text{root}(r))$, where r is the tree-to-tree translation rule and $\text{root}(r)$ is the root node pair corresponding to the rule.
(h4) Rule probability based on the source language, denoted as $\Pr(r|S_r)$, where r is the tree-to-tree translation rule and S_r is the source side of r.
(h5) Rule probability based on the target language, denoted as $\Pr(r|t_r)$, where r is the tree-to-tree translation rule and t_r is the target side of r.

Define the rule probability $\Pr(r)$. Here, we follow the definition of weighted synchronization grammar and use a log-linear model to give a score for each rule r as follows:

$$w(r) = \prod_{i} \theta_i(r)^{\lambda_i}, \tag{4}$$

where $\theta_i(r)$ is the ith feature defined on rule r and λ_i represents the weight of the ith feature. The feature set defined on each rule includes the features h1–h5 described above.

Consideration of the above five characteristics and their composition rating is hence expressed as follows:

$$\begin{aligned} w(r) = {} & \Pr(\overline{t_r}|\overline{s_r})^{\lambda_1} \times \Pr(\overline{s_r}|\overline{t_r})^{\lambda_2} \\ & \times \Pr_{lex}(\overline{t_r}|\overline{s_r}, a)^{\lambda_3} \times \Pr_{lex}(\overline{s_r}|\overline{t_r}, a)^{\lambda_4} \\ & \times \Pr(r|root(r))^{\lambda_5} \times \Pr(r|s_r)^{\lambda_6} \times \Pr(r|t_r)^{\lambda_7}. \end{aligned} \tag{5}$$

But because the corpus is small and the language characteristics play an important role in our scoring, we create a weight as a reward for the minimum rules, rule combinations, or rule generalizations that conform to the linguistic characteristics. These rewards can effectively improve the probability values of the rules conforming to language characteristics; as a result, in the decoding phase, such rules are more likely to be selected as candidate translations.

The three rewards that further consider the Vietnamese grammar are as follows:

(h6) Attributive postposition reward feature $AD(d)$.
(h7) Time adverbial postposition reward feature $TD(d)$.
(h8) Locative adverbial postposition reward feature $LD(d)$.

Thus, the final derived rating can be defined as follows:

$$w(d) = \prod_{r\in d} w(r) \times \exp(\lambda_{AD} \cdot AD(d)) \times \exp(\lambda_{TD} \cdot TD(d)) \times \exp(\lambda_{LD} \cdot LD(d)). \tag{6}$$

Normalizing Equation (6), we obtain

$$\Pr(d) = \frac{w(d)}{\sum_{d'\in D(S,T)} w(d')}, \tag{7}$$

where $D(S,T)$ represents all derivations that transform the source language tree S into the target language tree T.

3.2.2 Parameter Training. To acquire tree-to-tree translation rules, we need to train the model parameters. This training includes two aspects: (i) estimating the eigenvalues, that is, the values of the features defined in Section 3.2.1; and (ii) training the feature weight, that is, the optimization of the corresponding weights of the features.

For h1–h5, we use maximum likelihood estimation to estimate their eigenvalues. The idea of this approach is to adjust the parameter values to maximize the likelihood of the entire training set. The commonly used method is to continuously iterate through learning model parameters and update the expected count of each rule until a certain convergence condition is reached. Here, we use a simpler method of estimating parameters based on relative frequency. Theoretically, this method is equivalent to a maximum likelihood estimate of the model parameters when the desired frequency of 1 is given to each rule. For h1, we use the following equation to compute its probability value as

$$\Pr(\overline{t_r}|\overline{s_r}) = \frac{\sum_{r':\overline{s_{r'}}=\overline{s_r},\overline{t_{r'}}=\overline{t_r}} c(r')}{\sum_{r'':\overline{s_{r''}}=\overline{s_r}} c(r'')}. \tag{8}$$

This function represents the number of times the rule appears throughout the training set. The denominator represents the sum of the number of occurrences of all rules with the same source-language leaf-node sequence, and the numerator represents the sum of all the occurrences of the rules with the same source and target leaf-node sequences. Similarly, we can also estimate the value of h3–h5, respectively, as follows:

$$\Pr(r|root(r)) = \frac{c(r)}{\sum_{r':root(r')=root(r)} c(r')}, \tag{9}$$

$$\Pr(r|s_r) = \frac{c(r)}{\sum_{r':s_{r'}=s_r} c(r')}, \tag{10}$$

$$\Pr(r|t_r) = \frac{c(r)}{\sum_{r':t_{r'}=t_r} c(r')}. \tag{11}$$

We can also use this method to estimate the parameters $w(t_i|s_j)$ in h2:

$$\mathrm{w}(t_i|s_j) = \frac{\mathrm{c}(s_j,t_i)}{c(s_j)}. \tag{12}$$

Here, $C(s_j,t_i)$ represents the number of times that source language words s_j and target words t_i are aligned in the training corpus. $C(s_j)$ indicates the number of occurrences of source language words s_j in the training corpus.

To train the feature weights, we use the minimum error rate training proposed by Reference [28]. The core idea is to measure the number of errors in a sentence by comparing it with a reference sentence and minimize the error count. The optimization criterion is as follows:

$$\begin{aligned}\hat{\lambda}_1^M &= \arg\min_{\lambda_1^M}\left\{\sum_{s=1}^{S} E\left(r_s,\hat{e}\left(f_s;\lambda_1^M\right)\right)\right\} \\ &= \arg\min_{\lambda_1^M}\left\{\sum_{s=1}^{S}\sum_{k=1}^{K} E\left(r_s,e_{s,k}\right)\delta\left(\hat{e}\left(f_s;\lambda_1^M\right),e_{s,k}\right)\right\}\end{aligned}, \tag{13}$$

where function $\hat{e}$ is defined as

$$\hat{e}\left(f_s;\lambda_1^M\right) = \arg\max_{e\in C_s}\left\{\sum_{m=1}^{M}\lambda_m h_m(e|f_s)\right\}. \tag{14}$$

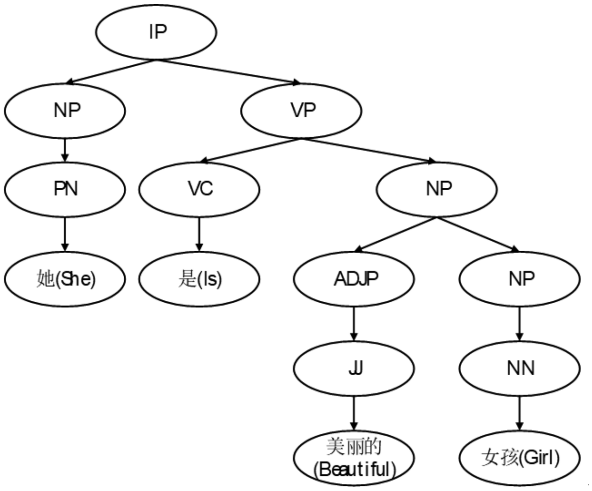

Fig. 3. Example of Chinese phrase syntax tree.

3.3 Decoding and Pruning

3.3.1 Decoding. The decoding process uses a tree-based analysis algorithm. From one (or more than one) source language syntax tree, the algorithm generates the target syntax tree of the optimal translation. In short, when the target tree is completed, the decoding process ends. The model is defined as follows:

$$t' = \arg\max_{t} \max_{d \in D(s,t)} \Pr(d), \tag{15}$$

where $D(s, t)$ represents all derivations that transform input sentence s into an estimation of the target sentence t.

Using the extracted syntax rules, all rules that can match the source language tree are associated with the corresponding source tree node and executed until the target tree has been constructed.

The decoding process is divided into the following steps:

(1) Input a syntax tree containing the syntax of the source language (the tree is actually composed of multiple sub trees).
(2) Each sub tree is a fragment that is matched with a rule in the rule library (a rule that has been matched in the rule library is flagged to avoid repeated access) until the tree is matched.
(3) Mark each solution as a new translation path and calculate its probability.
(4) Output the translation with the highest translation probability.

An example is as follows: for the Chinese sentence: "她是一个美丽的女孩 (She is a beautiful girl)," the Vietnamese translation is "Cô là" "她是 (She is)"; "cô gái" "女孩 (girl)"; "xinh đẹp" "美丽的 (beautiful)." The Chinese phrase syntax tree is shown in Figure 3.

We use the following extracted rules to translate the Chinese syntax tree:

IP/IP → NP VP | NP VP
PN/NP → "她 (She)" |Cô
VP/VP → VC "是 (Is)" NP | VC(là) NP
NP/NP → ADJP NN "女孩 (girl)" | NN(cô gái) ADJP
ADJP /ADJP → JJ "美丽的 (beautiful)" | JJ(xinh đẹp)

Table 3. Examples of Extracted Chinese-Vietnamese Syntax Rules

Extracted syntax rules
PN/NP → “她 (She)”/ anh ấy
PN/NP → “她 (She)”/ tôi
VP/VP → VC “是 (Is)” NP | VC(làm) NP
VP/VP → VC “是 (Is)” NP | VC(nghĩ) NP
NP/NP → ADJP NN “女孩 (girl)” | ADJP NN(đàn bà)
NP/NP → ADJP NN “女孩 (girl)” | NN(thiếu nữ)
ADJP /ADJP → JJ “美丽的 (beautiful)” | JJ(đẹp)
ADJP /ADJP → JJ “美丽的 (beautiful)” | JJ(Người đẹp)

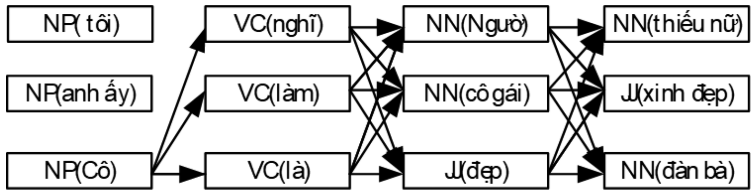

Fig. 4. Example search for candidate translations.

The translation rules are perfectly matched, but there are several candidate rules in the actual operation, as shown in Table 3 (only some of the candidate rules are listed).

For each sequence, there are generally more rules that can be used, and each available rule can generate a new search path. If the syntax rules contain unary non-lexical rules, then it must be considered whether the rule is reachable from the current search path.

3.3.2 Pruning with Language Feature. We use the matching rules to decode and the bottom-up beam search (beam search) method to complete the process. In general, tree parsing still needs to traverse all possible derivations. Intuitively, the number of candidate rules grows exponentially with the number of rules. The computational complexity of the decoding algorithm is expressed as follows:

$$\mathrm{O}(\max stack\ size * translation\ options\ ^{*}\ sentencelength) \tag{16}$$

The combination of rules can lead to an unsolvable problem space. Hence, we need to prune the search space of the search algorithm. Threshold pruning uses a fixed threshold value α that determines whether the translation hypothesis has been preserved by comparing the difference between α and the optimal candidate translation in a stack. If the translation hypothesis scores less than the score of the optimal hypothesis by a factor of α, then the translation hypothesis is pruned.

The effect of different thresholds α on the computational complexity of decoding is difficult to estimate. Therefore, we optimize the pruning of the candidate translations using language features. In the second decoding step, the two pairs of candidate translations are combined. When it is found that the language does not meet the definition in Equation (1), the combination of the rules will not form a path. In this way, the search space is greatly reduced and the decoding performance and accuracy are improved.

For example, the sentence “She is a beautiful girl” is shown in Figure 4. The process in the figure uses each of the rules to generate three candidate translations.

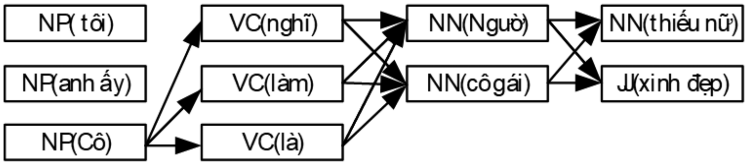

Fig. 5. Pruned candidate rules.

As Figure 4 shows, only $1 \times 3 \times 3 \times 3 = 27$ candidates are found for the translation of NP (Cô). We suppose that each rule has three candidate translations, giving a total of $3 \times 3 \times 3 \times 3 = 81$. The translation results of bundle search exponentially increase according to the increase in the candidate rules. It is not difficult to find by analyzing the sentences that one can use the attributive postposition rule to prune the candidate translations. Attributive postposition is defined as follows: NN ADJP (noun adjective, NN JJ). It is found that the result of the JJ NN translation in the bundle search does not conform to the syntax of Vietnamese. Hence, the JJ NN translation is removed as a candidate. The pruned rules are shown in Figure 5.

The number of candidate translations was reduced from 81 to $3 \times 3 \times 2 \times 2 = 36$. This case demonstrates that by deleting just two candidate translations, the number of translations can be reduced by more than half. Using the language-feature pruning, we not only reduce the search space but also improve the search efficiency. The best translation result is the one with the maximum probability after the probability of each segment is determined by an accumulative beam search.

4 EXPERIMENTS

4.1 Experimental Data

Because there is no public Chinese-Vietnamese bilingual sentence-based corpus, the experiment in this study used a Chinese-Vietnamese corpus collected from Chinese-Vietnamese parallel news articles on the Internet. The corpus was obtained using computer-based sentence alignment and manual proofing. At present, this collection of Chinese-Vietnamese parallel sentences includes over 100,000 sentences. The details of the experimental data are shown in Table 4 below.

4.2 Experimental Setup and Analysis

4.2.1 Chinese and Vietnamese Language Difference Statistics. To verify the bilingual differences between Chinese and Vietnamese summarized by the linguists described in Section 2 "Chinese-Vietnamese Bilingual Difference Analysis and Formalization," we randomly extracted out 10,000 sentence pairs from 80,000 Chinese-Vietnamese bilingual parallel sentence pairs to conduct statistics on syntactic differences between Chinese and Vietnamese. They include the appeared bilingual grammatical differences in the data set, the number of Chinese sentences and the number of Vietnamese sentences that contain grammatical difference structure, and the proportion of corresponding Vietnamese sentence numbers in the Chinese sentences. The results are shown in Table 5.

It can be seen from the statistical results in Table 5 that in the Chinese sentences containing the attributives, the proportion of the postpositions in the corresponding Vietnamese sentences accounts for more than 93%. Similarly, the proportion of time-adverbial postpositions in Vietnamese sentences is 88.3%, and the proportion of place-adverbial is 91%. A small number of temporal adverbials and place adverbials have no postposition.

4.2.2 Comparison of Different Rule Extraction Methods. First, the alignment quality of the combined rules was evaluated. To construct the evaluation data, 5,000 pairs of sentences were randomly

Table 4. Corpus for Experiments

Corpus		Chinese	Vietnamese
Training set	Sentences	80,000	
	Words	60,789	61,200
Development set	Sentences	10,000	
	Words	5,765	6,425
Language model training set	Sentences	80,000	
	Words	60,789	61,200
Test set	Sentences	5,000	
	Words	4,121	5,220

Table 5. The Chinese-Vietnamese Bilingual Difference Statistics

The Chinese-Vietnamese bilingual differences	Chinese sentences	Vietnamese sentences	Difference proportion
(1) Adverb (to modify verb) postposition	3,468	3,277	94.5%
(2) Adverb (to modify adjective) postposition	2,896	2,714	93.7%
(3) Adjective (to modify noun) postposition	6,832	6,758	98.9%
(4) Time-adverbial postposition	1,231	1,087	88.3%
(5) Place-adverbial postposition	958	872	91.0%

selected from the training bilingual parallel dataset (the average length of Chinese sentences was 20.2 words, while the average length of Vietnamese sentences was 23.8 words), and two markers manually label node alignment (the label consistency was higher than 80%). We used the alignment precision (P), recall rate (R), and F1 values to evaluate the quality of the combination rule alignment. These values are calculated respectively as follows:

$$P = \frac{\text{Rule number extracted properly}}{\text{Total rule number extracted}} \times 100\%, \tag{17}$$

$$R = \frac{\text{Rule number extracted properly}}{\text{Total rule number to be extracted}} \times 100\%, \tag{18}$$

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\%. \tag{19}$$

The following methods were compared:

Baseline: Normal tree-to-tree rule template library extraction method.

Baseline + LF: The abbreviated + LF, minimum syntax rule extraction method with language features.

Baseline + RG: The abbreviated + RG, rule extraction method with rule generalization.

Baseline + RG + RC: The abbreviated + RG + RC, rule extraction method with rule generalization and rule combination.

Baseline + LF + RG + RC: The abbreviated + LF + RG + RC, rule extraction method with language features, rule generalization, rule combination.

Table 6 shows the results of these alignment approaches.

The language features and rule combination approaches have a distinct advantage over the baseline system. This is mainly because the proposed method significantly improves the recall rate of the alignment results, which also improves the F1 value substantially.

Table 6. Performance of Different Rule Extraction Methods

Rule extraction method	P (%)	R (%)	F1 (%)
Baseline	64.8	60.9	62.9
Baseline + LF	68.4	70	64
Baseline + RG	66.8	62.6	63
Baseline + RG + RC	70.2	68.5	69.2
Baseline + LF + RG + RC	70	71.2	72.3

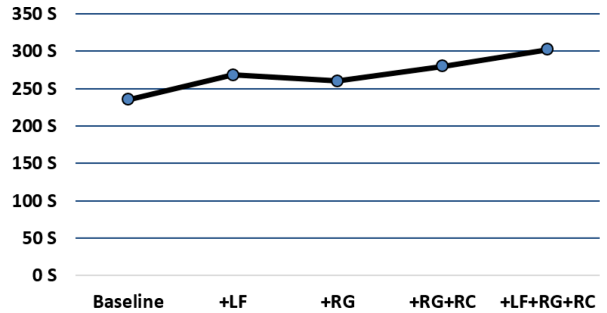

Fig. 6. Efficiency of different extraction methods.

Figure 6 shows the efficiency of the various extraction methods. The results show that the language feature, rule combination, and rule generalization approaches have some influence on the performance of the system. Our method with language feature, rule combination, and rule generalization is about 1min slower than the baseline system when extracting syntax rules for 5,000 sentence pairs. Its runtime does not show exponential growth with respect to baseline system, so we conclude that this approach improves performance of extracting rules overall.

4.2.3 Performance Evaluation on Vietnamese Phrase Syntax Parsing. To evaluate the performance of different phrase syntax parsers for Vietnamese, a self-developed and improved PCFG Vietnamese phrase tree parser with language features [29] is compared with a PCFG-based Vietnamese phrase tree parser and a Vietnamese phrase tree parser based on maximum entropy.

The experimental corpus contains 10,000 Vietnamese phrase trees from the Pennsylvania Tree Library, of which 8,000 are used as training corpus and 2,000 sentences as test corpus. The latter is divided into four closed test sets of 500 sentences. 25,981 Vietnamese sentences are obtained from Internet news, blogs, forums, and so on. After tagging part-of-speech on them, four 100-sentence open test sets from different categories are selected. The PCFG-based Vietnamese phrase tree analysis method is referred to as the PCFG method.

The Vietnamese phrase tree analysis method of combining improved PCFG with language features is referred to as the LG+PCFG method.

The Vietnamese syntactic analysis method based on maximum entropy is referred to as the Max-entropy method.

In the models training from the 8,000-sentence Pennsylvania phrase tree as training corpus, the Vietnamese grammar rule set is counted as the initial grammatical probability set. On them, the PCFG model and the Max-entropy model are trained. Then, from the 125,981 Vietnamese sentences, the Vietnamese language-feature set is counted as a supplement to the initial grammatical probability set. On them, the LG+PCFG model is trained.

Table 7. Performance Evaluation on F-Value from Different Vietnamese Syntax Parsers

Method	Test set 1	Test set 2	Test set 3	Test set 4	Test set 5	Test set 6	Test set 7	Test set 8
PCFG	77.34	76.49	77.39	77.56	73.17	74.60	72.29	73.61
Max-entropy	79.56	78.31	79.22	80.43	75.19	76.68	75.23	75.89
LG + PCFG	83.89	81.49	83.39	81.56	77.17	79.60	76.29	78.61

Table 8. Rule Base Sizes from Different Translation Methods

Method	Rule library size (MB)	Number of rules in the rule base (millions)
Baseline 1	482	3.16
Baseline 2	300	2.49
Baseline 2 + LF + RG + RC	356	2.84

On the eight test sets, we compare F values of Vietnamese phrase tree parsing by the PCFG method, the Max-entropy method, and the LG+PCFG method. The experimental results are shown in Table 7.

As can be seen from Table 7, the F-value of the Max-entropy method is higher than that of the conventional PCFG method on the eight test sets, and the F value of the LG+PCFG method is higher than that of the conventional PCFG method and the Max-entropy method value, mainly because its fusion of Vietnamese language feature sets played a positive role in syntactic tree analysis.

4.2.4 Comparison of Translation Models. The experimental platform for this evaluation uses the NiuTrans.SMT system developed by Northeastern University [15]. For the Chinese syntax parsing, we use the Stanford Parser. The Vietnamese tree parsing is based on the Vietnamese phrase treebank construction method presented by Reference [29]. The precision of this parser can reach 80.09%. We used GIZA++ to obtain the Chinese-Vietnamese word alignment results. The ternary language model was trained by an open-source machine-translation platform. The minimum error rate training described in Section 3.2 (Equation (13)) was used to train the parameters. We used the BLEU evaluation criterion as the evaluation metric.

The following methods were compared on rule size (see Table 8), BLEU-value of translation, and runtime of training and decoding:

Baseline 1: Chinese-Vietnamese translation method based on hierarchical phrase translation model.

Baseline 2: Chinese-Vietnamese tree-to-tree translation model based on syntax.

Baseline 3: The abbreviated NMT, neural network-based machine translation with an attention mechanism [3]. We used a four-layer network in the experiment. The number of GRUs was 1,024, and the number of training steps was 340,000.

Baseline 2 + LF + RG + RC: The abbreviated + LF + RG + RC, Chinese-Vietnamese tree-to-tree translation model based on linguistic features, rule generalization, and rule combination.

Baseline 2 + PR: The abbreviated + PR, Chinese-Vietnamese translation method based on tree-to-tree when decoding with language features.

Our method: Baseline 2 + LF + RG + RC + PR.

For the size of the rule base extracted by different translation methods, the results are shown in Table 8. They indicate that the rule base of Baseline 1 is the largest, followed by our method Baseline 2 + LF + RG + RC. The Baseline 2 translation model has the smallest rule base. Our method expands the size of the rule base, covering more language features.

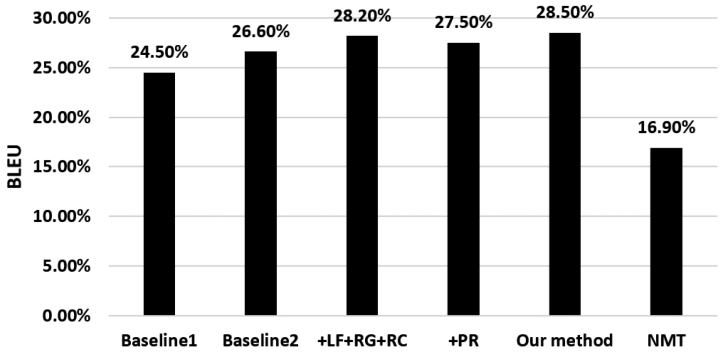

Fig. 7. Performance for different translation methods.

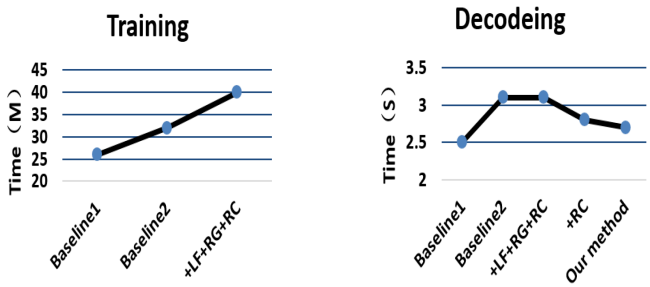

Fig. 8. Efficiency of different translation methods.

For the performance evaluation of different translation methods, we use the BLEU-value. Figure 7 shows the experimental results. Because very little grammatical knowledge is contained in Baseline 1, its BLEU reaches only 24.50 and leads to the worst performance. Moreover, Baseline 2, although fusing syntactic information, rises up only +2.1 BLEU. For Baseline 2 + LF + RG + RC, the first method proposed in the article, integrating language features, combination rules, and generalization rules, the BLEU-value is rapidly increased to 28.2. Analysis found that the method resolves some language features uncovered by the rule base and deletes some rules that do not conform to the pre-defined syntax characteristics. It addresses both the high error rate and low coverage problems. The proposed method shows good performance, increasing the BLEU by 1.6% with respect to Baseline 2. Baseline 2 + PR the second method proposed in the article, does pruning using language features when decoding, reduces the search space, and hence reduces the error rate. It therefore has a higher BLEU value than that of Baseline 2 by 0.9%. For Baseline 2 + LF + RG + RC + PR, our method, because of integrating all features mentioned above, has BLEU reach to 28.50, and its performance improvement is most obvious. As for the NMT model, although it can generate a more human-like translation, it also requires a larger corpus. Therefore, this method obtained a rather low BLEU value with small corpus in this experiment. According to an experiment we performed on a Chinese-English corpus, its performance would be comparable to that of SMT performance only when given a much larger corpus.

For the training runtime and decoding runtime of different translation methods, the experimental results are shown in Figure 8.

From Figure 8, it can be seen that in training phase, compared to Baseline 1, Baseline 2 is slower by less than 10mins, while Baseline 2 + LF + RG + RC is also slower by an additional 10mins compared to Baseline 2. In decoding phase, when translating a single sentence,

Table 9. Example 1 of Different Machine Translation Methods

Chinese	她是一个非常美丽的女孩 (She is a very beautiful girl.)
Vietnamese from Baseline1	Cô (She) là (is) một (a) đẹp (beautiful) cô gái (girl).
Vietnamese from Baseline2	Cô (She) là (is) một (a) cô gái (girl) xinh đẹp (beautiful).
Vietnamese from NMT	Cô (She) là (is) một (a) cô gái (girl) rất (very) xinh đẹp (beautiful).
Vietnamese from Baseline 2 + LF + RG + RC	Cô (She) là (is) một (a) cô gái (girl) xinh đẹp (beautiful) rất (very).
Vietnamese from Baseline 2 + LF + RG + RC + PR	Cô ấy (She) là (is) một (a) cô gái (girl) xinh đẹp (beautiful) rất (very).
Reference translation 1	cô ấy (She) là(is) một (a) cô gái (girl) xinh đẹp (beautiful) rất (very).
Reference translation 2	Cô ấy (She) là(is) một (a) cô gái (girl) xinh đẹp (beautiful) cực kỳ (very).

Baseline 2 + LF + RG + RC + PR, with language features, rule generalizations, rule combinations, and pruning in the decoding with language features, expends 2.62 seconds on decoding—faster than Baseline 2 by 0.52s and clearly improved the efficiency. It is only slightly slower than Baseline 1. In general, the proposed system improves the performance and guarantees the efficiency.

For the Chinese sentence "她是一个非常美丽的女孩 (She is a very beautiful girl)" and the Vietnamese sentence "Hiện Cơ quan Cảnh sát điều tra, Bộ Công an đang khẩn trương điều tra, thu hồi tài sản để xử lý theo đúng quy định của pháp luật," translation is carried out by Baseline 1, Baseline 2, NMT, Baseline 2 + LF + RG + RC, and Baseline 2 + LF + RG + RC + PR. The translation results are shown in Table 9 and Table 10, respectively:

Compare the five target Vietnamese sentences from Table 9 and find that the translation results of Baseline 2 + LF + RG + RC and Baseline 2 + LF + RG + RC + PR are better than those of Baseline 1 and Baseline 2. Obviously, it is syntactic information, rule generalization and bilingual language differences that enhance post-characteristics of modifier word "rất" and make translation accuracy improve greatly. After syntactic information and bilingual language differences are integrated into our translation model, the leakage problem in neural machine translation is fully solved, and the target sentence is more complete and more consistent with the source sentence.

It can be seen from Table 10 that Chinese from Baseline 2 + LF + RG + RC + PR, with a fusion of language feature, rule generalization, rule combination, and decoding optimization, gains better accuracy than the other methods in the Vietnamese-to-Chinese translation task.

4.2.5 Influence of Different Vietnamese Syntax Parsing Method on Translation Performance. To analyze how the accuracy of the Vietnamese phrase tree parsing influences the translation results from the proposed translation model, on the same training corpus and test corpus, we use the traditional PCFG model, the maximum entropy model, and the improved PCFG model with language feature to parse the Vietnamese phrase tree. Then, the traditional syntactic tree-to-tree translation model (Baseline 2) and the proposed tree-to-tree syntax statistical machine-translation model (Baseline 2 + LF + RG + RC + PR) are compared in the translating experiment. The BLEU values are shown in Table 11.

Table 10. Example 2 of Different Machine Translation Methods

Chinese	目前，警察调查局，公安部正在开展紧急调查，依据正确的法律规定追回财产处理。(At present, the Police Investigation Bureau, the Ministry of Public Security is conducting an emergency investigation and recovering the property in accordance with the correct legal provisions.)
Vietnamese from Baseline1	Hiện tại (At present), Cảnh sát (Cop) Phòng điều tra (Investigation Department), Bộ Công (Ministry of Public Security) đang được tiến hành (is underway) an khẩn cấp (an emergency) Một cuộc điều tra (An inquiry), Cơ sở (basis) Đúng (correct) Luật (legal) Quy định (regulations) Phục hồi (rehibilitate) Tài sản (asset) Chế biến (processing).
Vietnamese from Baseline2	Hiện tại (At present), Cục điều tra công an (Police Investigation Department), Bộ Công (Ministry of Public Security) Một cuộc điều tra (An investigation) khẩn cấp (urgent) đang được tiến hành (is underway), Cơ sở (basis) Đúng (correct) Luật (legal) Quy định (regulations) Phục hồi (rehibilitate) Tài sản (asset) Chế biến (processing).
Vietnamese from NMT	Hiện nay (Currently), cơ quan công (the police) an đang triển khai điều tra (is under investigation) khẩn cấp (emergency), bộ công an (police) đang truy thù xử lý (retaliate processing) tài sản (the property) theo quy định pháp luật đúng (in accordance with the law).
Vietnamese from Baseline 2 + LF + RG + RC	Hiện tại (Currently), Phòng điều(Investigation Agency) tra Cảnh sát (the Police), Bộ Công an(Ministry of Public Security) Thực hiện(Carry out) an đang(Being) Điều tra khẩn cấp(Emergency investigation) Khôi phục việc xử lý(recovering) tài sản(property) theo đúng các quy định(in accordance with) pháp lý(law) chính xác(regulations).
Vietnamese from Baseline 2 + LF + RG + RC + PR	Hiện tại (Currently), Trạm cảnh sát (the police Bureau) sát Phòng điều(Investigation Agency), Bộ Công an(Ministry of Public Security) Đang (Being)thực hiện(carry out) Điều tra khẩn cấp(Emergency investigation) Khôi phục việc xử lý(recovering) tài sản(property) theo đúng các quy định(in accordance with) pháp lý(law) chính xác(provisions).
Reference translation 1	Trạm cảnh sát (the police Bureau) điều tra (investigating), Bộ Công an(Ministry of Public Security) Đang (Being) mạnh mẽ (intensively) tiến hành điều tra (conducting investigations) xử lý(handling) tài sản(property) theo đúng các quy định(in accordance with) đúng quy định của pháp luật(correct legal provisions)
Reference translation 2	Hiện tại (Currently), Cảnh sát(police) Bộ Công an(Ministry of Public Security) khẩn trương điều tra(Urgent investigation), theo đúng các quy định(in accordance with) đúng quy định của pháp luật(correct legal provisions)

As can be seen from Table 11, whether in the traditional syntax tree-to-tree translation method or in the proposed tree-to-tree syntax machine-translation method, the BLEU with the LG + PCFG parser is higher than that of the PCFG parser and higher than that of the max-entropy parser, reaching 26.6 and 28.5, respectively, which fully demonstrates that the parsing accuracy of the Vietnamese phrase tree has a positive contribution on the Chinese-Vietnamese tree-to-tree syntax machine-translation method. However, for the proposed translation method (Baseline 2 + LF + RG + RC + PR), because the bilingual language differences have been incorporated into the translation model, the performance of Vietnamese syntax parsing has little effect on the BLEU of translation candidates. The BLEU with the LG + PCFG syntax parser only get a +0.9 rise on the basis of the PCFG syntax parser. In addition, we can see from Table 11 that when using the

Table 11. BLEU-values of Machine Translation Methods with Different Vietnamese Syntax Parsers

Vietnamese statement parser	BLEU-value of the traditional syntax tree to tree machine-translation method (Baseline 2)	BLEU-value of the proposed tree-to-tree syntax machine-translation method (Baseline 2 + LF + RG + RC + PR)
PCFG	21.8	27.6
Max-entropy	23.7	27.9
LG+PCFG	26.6	28.5

LG + PCFG method to parse the Vietnamese phrase tree, the BLEU proposed in this article gets a +1.9 rise on the basis of the traditional tree-to-tree syntax machine-translation method. The reason is that rule generalization, rule combination, and decoding with language features have positive contributions on performance of the proposed Baseline 2 + LF + RG + RC + PR translation method.

5 CONCLUSIONS

To address sparse data problems that will occur when only small corpora (such as Chinese-Vietnamese corpora) are available, this article proposed a Chinese-Vietnamese tree-to-tree syntax statistical translation model with language difference. It makes full use of bilingual syntax information as well as formal language reward rules and features to reduce the error rate of rule extraction. And it improves the size and coverage of rule base through rule combination and rule generalization. In the decoding phase, it introduces language features to make pruning optimization. As a result, it reduces the decoding search space and improves the accuracy of candidate translations. The experimental results show that the proposed model outperforms the traditional tree-to-tree translation model while retaining efficiency. It also shows that language-difference features have a good supervised effect on the Chinese-Vietnamese translation task.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the reviewers' insightful comments, which helped improve the manuscript.

REFERENCES

[1] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1700–1709.

[2] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Advances in Neural Information Processing Systems Conference*. 3104–3112.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. Retrieved from arXiv preprint arXiv:1409.0473.

[4] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. Retrieved from arXiv preprint arXiv:1711.00043.

[5] Jinsong Su, Jiali Zeng, Deyi Xiong, Yang Liu, Mingxuan Wang, and Jun Xie. 2018. A hierarchy-to-sequence attentional neural machine translation model. *IEEE/ACM Trans. Aud., Speech Lang. Proc.* 26, 3 (2018), 623–632.

[6] Yun Chen, Yang Liu, and Victor O. K. Li. 2018. Zero-resource neural machine translation with multi-agent communication game. Retrieved from arXiv preprint arXiv:1802.03116.

[7] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. Retrieved from arXiv preprint arXiv:1609.08144.

[8] David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 263–270.

[9] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 961–968.

[10] Jiajun Zhang, Feifei Zhai, and Chengqing Zong. 2011. Augmenting string-to-tree translation models with fuzzy use of source-side syntax. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 204–215.

[11] Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 609–616.

[12] Graham Neubig and Kevin Duh. 2014. On the elements of an accurate tree-to-string machine translation system. In *Proceedings of the 52nd Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 143–149.

[13] Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proceedings of the Meeting of the Association for Computational Linguistics: Human Language Technologies*. 559–567.

[14] Tong Xiao and Jingbo Zhu. 2013. Unsupervised sub-tree alignment for tree-to-tree translation. *J. Artific. Intell. Res.* 48 (2013), 733–782.

[15] Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. NiuTrans: An open source toolkit for phrase-based and syntax-based machine translation. In *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 19–24.

[16] Tong Xiao. 2012. *On Learning and Decoding Approaches to Tree-to-Tree Statistical Machine Translation*. Ph.D. Dissertation. Northeastern University, Shenyang, China.

[17] Phuoc Tran, Dien Dinh, and Hien T. Nguyen. 2016. A character level based and word level based approach for Chinese-Vietnamese machine translation. *Computat. Intell. Neurosci.* 2016 (2016), 9821608.

[18] Phuoc Tran, Dien Dinh, and Linh Tran. 2014. Resolving named entity unknown word in Chinese-Vietnamese machine translation. *Adv. Intell. Syst. Comput.* 245 (2014), 273–284.

[19] Phuoc Tran, Dien Dinh, Tan Le, and Thao Nguyen. 2013. Handling organization name unknown word in Chinese-Vietnamese machine translation. In *Proceedings of the IEEE-RIVF International Conference on Computing & Communication Technologies*.

[20] Jianyalin He, Zhengtao Yu, Changtao Lv, Hua Lai, Shengxiang Gao, and Yang Zhang. 2017. Language post positioned characteristic based Chinese-Vietnamese statistical machine translation method. In *Proceedings of the International Conference on Asian Language Processing*. IEEE, 180–184.

[21] Hai Zhao, Tianjiao Yin, and Jingyi Zhang. 2013. Vietnamese to Chinese machine translation via Chinese character as pivot. In *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation*. 250–259.

[22] Vu Thi Ha. 2005. A comparison between Vietnamese and Chinese syntactic constituent orders. *J. Yunnan Norm. Univ.* 3, 6 (2005), 65–68.

[23] Haitao Mi, Huang Liang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of the Meeting of the Association for Computational Linguistics: Human Language Technologies*. 192–199.

[24] Hao Zhang, Licheng Fang, Peng Xu, and Xiaoyun Wu. 2011. Binarized forest to string translation. In *Proceedings of the 49th Meeting of the Association for Computational Linguistics: Human Language Technologies—Volume 1*. Association for Computational Linguistics, 835–845.

[25] Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 138–141.

[26] Zhongqiang Huang, Martin Čmejrek, and Bowen Zhou. 2010. Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 138–147.

[27] Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 44–52.

[28] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Meeting on Association for Computational Linguistics—Volume 1*. Association for Computational Linguistics, 160–167.

[29] Ying Li, Jianyi Guo, Zhengtao Yu, Yantuan Xian, and Yonghua Wen. 2016. Building the Vietnamese phrase treebank by improved probabilistic context-free grammars. In *Proceedings of the China Workshop on Machine Translation*. Springer, 75–90.

Received December 2017; revised December 2018; accepted January 2019