A Continuous Space Rule Selection Model for Syntax-based Statistical Machine Translation

Jingyi Zhang[1,2], **Masao Utiyama**[1], **Eiichro Sumita**[1]
Graham Neubig[2], **Satoshi Nakamura**[2]
[1]National Institute of Information and Communications Technology,
3-5Hikaridai, Keihanna Science City, Kyoto 619-0289, Japan
[2]Graduate School of Information Science, Nara Institute of Science and Technology,
Takayama, Ikoma, Nara 630-0192, Japan
jingyizhang/mutiyama/eiichiro.sumita@nict.go.jp
neubig/s-nakamura@is.naist.jp

Abstract

One of the major challenges for statistical machine translation (SMT) is to choose the appropriate translation rules based on the sentence context. This paper proposes a continuous space rule selection (CSRS) model for syntax-based SMT to perform this context-dependent rule selection. In contrast to existing maximum entropy based rule selection (MERS) models, which use discrete representations of words as features, the CSRS model is learned by a feed-forward neural network and uses real-valued vector representations of words, allowing for better generalization. In addition, we propose a method to train the rule selection models only on minimal rules, which are more frequent and have richer training data compared to non-minimal rules. We tested our model on different translation tasks and the CSRS model outperformed a baseline without rule selection and the previous MERS model by up to 2.2 and 1.1 points of BLEU score respectively.

1 Introduction

In syntax-based statistical machine translation (SMT), especially tree-to-string (Liu et al., 2006; Graehl and Knight, 2004) and forest-to-string (Mi et al., 2008) SMT, a source tree or forest is used as input and translated by a series of tree-based translation rules into a target sentence. A tree-based translation rule can perform reordering and translation jointly by projecting a source subtree into a target string, which can contain both terminals and nonterminals.

One of the difficulties in applying this model is the ambiguity existing in translation rules: a source subtree can have different target translations extracted from the parallel corpus as shown in Figure 1. Selecting correct rules during decoding is a major challenge for SMT in general, and syntax-based models are no exception.

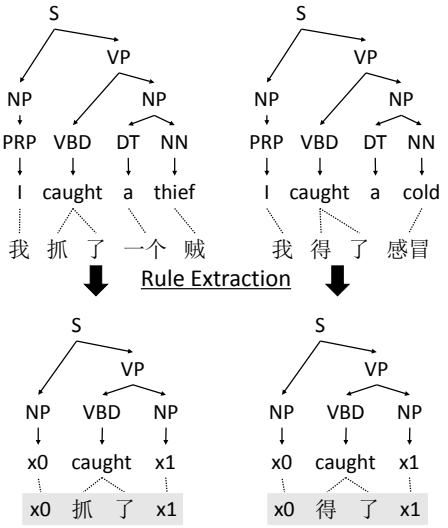

Figure 1: An ambiguous source subtree with different translations (English-to-Chinese).

There have been several methods proposed to resolve this ambiguity. The most simple method, used in the first models of tree-to-string translation (Liu et al., 2006), estimated the probability of a translation rule by relative frequencies. For example, in Figure 1, the rule that occurs more times in the training data will have a higher score. Later, Liu et al. (2008) proposed a maximum entropy based rule selection (MERS, Section 2) model for syntax-based SMT, which used contextual information for rule selection, such as words surrounding a rule and words covered by nonterminals in a rule. For example, to choose the correct rule from the two rules in Figure 1 for decoding a particular input sentence, if the source phrase covered by "x1" is "a thief" and this child phrase

has been seen in the training data, then the MERS model can use this information to determine that the first rule should be applied. However, if the source phrase covered by "x1" is a slightly different phrase, such as "a gunman", it will be hard for the MERS model to select the correct rule, because it treats "thief" and "gunman" as two different and unrelated words.

In this paper, we propose a continuous space rule selection (CSRS, Section 3) model, which is learned by a feed-forward neural network and replaces the discrete representations of words used in the MERS model with real-valued vector representations of words for better generalization. For example, the CSRS model can use the similarity of word representations for "gunman" and "thief" to infer that "a gunman" is more similar with "a thief" than "a cold".

In addition, we propose a new method, applicable to both the MERS and CSRS models, to train rule selection models only on minimal rules. These minimal rules are more frequent and have richer training data compared to non-minimal rules, making it possible to further relieve the data sparsity problem.

In experiments (Section 4), we validate the proposed CSRS model and the minimal rule training method on English-to-German, English-to-French, English-to-Chinese and English-to-Japanese translation tasks.

2 Tree-to-String SMT and MERS

2.1 Tree-to-String SMT

In tree-to-string SMT (Liu et al., 2006), a parse tree for the source sentence F is transformed into a target sentence E using translation rules R. Each tree-based translation rule $r \in R$ translates a source subtree $\tilde{t}$ into a target string $\tilde{e}$, which can contain both terminals and nonterminals. During decoding, the translation system examines different derivations for each source sentence and outputs the one with the highest probability,

$$\hat{E} = \arg\max_{E,R} \Pr(E, R|F). \quad (1)$$

For a translation E of a source sentence F with derivation R, the translation probability is calculated as follows,

$$\Pr(E, R|F) \approx \frac{\exp\left(\sum_{k=1}^{K} \lambda_k h_k(E, R, F)\right)}{\sum_{E',R'} \exp\left(\sum_{k=1}^{K} \lambda_k h_k(E', R', F)\right)}. \quad (2)$$

Here, h_k are features used in the translation system and λ_k are feature weights. Features used in Liu et al. (2006)'s model contain a language model and simple features based on relative frequencies, which do not consider context information.

One of the most important features used in this model is based on the log conditional probability of the target string given the input source subtree $\log \Pr(\tilde{e}|\tilde{t})$. This allows the model to determine which target strings are more likely to be used in translation. However, as the correct translation of the rules may depend on context that is not directly included in the rule, this simple context-independent estimate is inherently inaccurate.

2.2 Maximum Entropy Based Rule Selection

To perform context-dependent rule selection, Liu et al. (2008) proposed the MERS model for syntax-based SMT. They built a maximum entropy classifier for each ambiguous source subtree $\tilde{t}$, which introduced contextual information C and estimated the conditional probability using a log-linear model as shown below,

$$\Pr(\tilde{e}|\tilde{t}, C) = \frac{\exp\left(\sum_{k=1}^{K} \lambda_k h_k(\tilde{e}, C)\right)}{\sum_{\tilde{e}'} \exp\left(\sum_{k=1}^{K} \lambda_k h_k(\tilde{e}', C)\right)}. \quad (3)$$

The target strings $\tilde{e}$ are treated as different classes for the classifier.

Supposing that,

- r covers source span $[f_\varphi, f_\vartheta]$ and target span $[e_\gamma, e_\sigma]$,
- $\tilde{t}$ contains K nonterminals $\{X_k | 0 \le k \le K-1\}$,
- X_k covers source span $[f_{\varphi_k}, f_{\vartheta_k}]$ and target span $[e_{\gamma_k}, e_{\sigma_k}]$,

the MERS model used 5 kinds of source-side features as follows,

1. Lexical features: words around a rule (e.g. $f_{\varphi-1}$) and words covered by nonterminals in a rule (e.g. f_{φ_0}).

2. Part-of-speech features: part-of-speech (POS) of context words that are used as lexical features.

3. Span features: span lengths of source phrases covered by nonterminals in r.

4. Parent features: the parent node of $\tilde{t}$ in the parse tree of the source sentence.

5. Sibling features: the siblings of the root of $\tilde{t}$.

Note that the MERS model does not use features of the source subtree $\tilde{t}$, because the source subtree $\tilde{t}$ is fixed for each classifier.

The MERS model was integrated into the translation system as two additional features in Equation 2. Supposing that the derivation R contains M rules $r_1, ..., r_M$ with ambiguous source subtrees, then these two MERS features are as follows,

$$\begin{aligned} h_1(E,R,F) &= \sum_{m=1}^{M} \log \Pr\left(\tilde{e}_m|\tilde{t}_m, C_m\right) \\ h_2(E,R,F) &= M, \end{aligned} \tag{4}$$

where $\tilde{t}_m$ and $\tilde{e}_m$ are the source subtree and the target string contained in r_m, and C_m is the context of r_m. h_1 is the MERS probability feature, and, h_2 is a penalty feature counting the number of predictions made by the MERS model.

3 Our CSRS Approach

3.1 Modeling

The proposed CSRS model differs from the MERS model in three ways.

1. Instead of learning a single classifier for each source subtree $\tilde{t}$, it learns a single classifier for all rules.

2. Instead of hand-crafted features, it uses a feed-forward neural network to induce features from context words.

3. Instead of one-hot representations, it uses distributed representations to exploit similarities between words.

First, with regard to training, our CSRS model follows Zhang et al. (2015) in approximating the posterior probability by a binary classifier as follows,

$$\Pr\left(\tilde{e}|\tilde{t}, C\right) \approx \Pr\left(v = 1|\tilde{e}, \tilde{t}, C\right), \tag{5}$$

where $v \in \{0, 1\}$ is an indicator of whether $\tilde{t}$ is translated into $\tilde{e}$. This is in contrast to the MERS model, which treated the rule selection problem as a multi-class classification task. If instead we attempted to estimate output probabilities for all different $\tilde{e}$, the cost of estimating the normalization coefficient would be prohibitive, as the number of unique output-side word strings $\tilde{e}$ is large. There are a number of remedies to this, including noise contrastive estimation (Vaswani et al., 2013), but the binary approximation method has been reported to have better performance (Zhang et al., 2015).

To learn this model, we use a feed-forward neural network with structure similar to neural network language models (Vaswani et al., 2013). The input of the neural rule selection model is a vector representation for $\tilde{t}$, another vector representation for $\tilde{e}$, and a set of ξ vector representations for both source-side and target-side context words of r:

$$C(r) = w_1, ..., w_\xi \tag{6}$$

In our model, $C(r)$ is calculated differently depending on the number of nonterminals included in the rule. Specifically, Equation 7 defines $C_{out}(r, n)$ to be context words (n-grams) around r and $C_{in}(r, n, X_k)$ to be boundary words (n-grams) covered by nonterminal X_k in r.[1]

$$\begin{aligned} &C_{out}(r,n) \\ &= f_{\varphi-n}^{\varphi-1}, f_{\theta+1}^{\theta+n}, e_{\gamma-n}^{\gamma-1}, e_{\sigma+1}^{\sigma+n} \\ &C_{in}(r,n,X_k) \\ &= f_{\varphi_k}^{\varphi_k+n-1}, f_{\theta_k-(n-1)}^{\theta_k}, e_{\gamma_k}^{\gamma_k+n-1}, e_{\sigma_k-(n-1)}^{\sigma_k} \end{aligned} \tag{7}$$

The context words used for a translation rule r with K nonterminals are shown as below.

K	$C(r)$
$=0$	$C_{out}(r,6)$
$=1$	$C_{out}(r,4), C_{in}(r,2,X_0)$
>1	$C_{out}(r,2), C_{in}(r,2,X_0), C_{in}(r,2,X_1)$

We can see that rules with different numbers of nonterminals K use different context words.[2]

[1] Note that when extracting C_{out}, we use "$\langle s \rangle$" and "$\langle /s \rangle$" for context words that exceed the length of the sentence; When extracting C_{in}, we use "$\langle non \rangle$" for context words that exceed the length of the nonterminal. Words that occur less than twice in the training data are replaced by "$\langle unk \rangle$".

[2] In most cases, restrictions on extracted rules will ensure that rules will only contain two nonterminals. However, when using minimal rules as described in the next section, more than two nonterminals are possible, and in these cases, only contextual information covered by the first two nonterminals is used in the input. These cases are sufficiently rare, however, that we chose to consider only the first two.

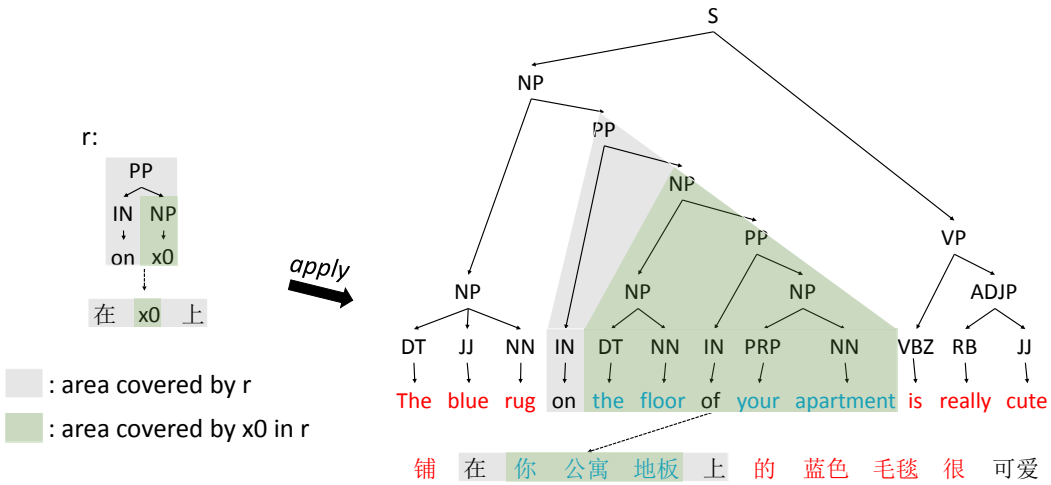

Figure 2: Context word examples. The red words are contained in $C_{out}(r, 4)$ and the blue words are contained in $C_{in}(r, 2, X_0)$.

For example, if r does not contain nonterminals, then C_{in} is not used. Besides, we use more context words surrounding the rule ($C_{out}(r, 6)$) for rules with $K = 0$ than rules that contain nonterminals ($C_{out}(r, 4)$ for $K = 1$ and $C_{out}(r, 2)$ for $K > 1$). This is based on the intuition that rules with $K = 0$ can only use the context words surrounding the rule as information for rule selection, hence this information is more important than for other rules. Figure 2 gives an example of context words when applying the rule r to the example sentence.

Note that we use target-side context because source-side context is not enough for selecting correct rules. Since it is not uncommon for one source sentence to have different correct translations, a translation rule used in one correct derivation may be incorrect for other derivations. In these cases, target-side context is useful for selecting appropriate translation rules. [3]

The vector representations for $\tilde{t}$, $\tilde{e}$ and C are obtained by using a projection matrix to project each one-hot input into a real-valued embedding vector. This projection is another key advantage over the MERS model. Because the CSRS model learns one unified model for all rules and can share all training data to learn better vector representations of words and rules, and the similarities between vectors can be used to generalize in cases such as the "thief/gunman" example in the introduction.

[3]It is also possible to consider target-side context in a framework like the MERS model, but we show in experiments that a linear model using the same features as the CSRS model did not improve accuracy.

After calculating the projections, two hidden layers are used to combine all inputs. Finally, the neural network has two outputs $\Pr(v = 1|\tilde{e}, \tilde{t}, C)$ and $\Pr(v = 0|\tilde{e}, \tilde{t}, C)$.

To train the CSRS model, we need both positive and negative training examples. Positive examples, $\langle \tilde{e}, \tilde{t}, C, 1 \rangle$, can be extracted directly from the parallel corpus. For each positive example, we generate one negative example, $\langle \tilde{e}', \tilde{t}, C, 0 \rangle$. Here, $\tilde{e}'$ is randomly generated according to the translation distribution (Zhang et al., 2015),

$$\Pr(\tilde{e}|\tilde{t}) = \frac{Count(\tilde{e}, \tilde{t})}{\sum_{\tilde{e}'} Count(\tilde{e}', \tilde{t})}, \quad (8)$$

where, $Count(\tilde{e}, \tilde{t})$ is how many times $\tilde{t}$ is translated into $\tilde{e}$ in the parallel corpus.

During translating, following the MERS model, the CSRS model only calculates probabilities for rules with ambiguous source subtrees. These predictions are converted into two CSRS features for the translation system similar to the two MERS features in Equation 4: one is the product of probabilities calculated by the CSRS model and the other one is a penalty feature that stands for how many rules with ambiguous source subtrees are contained in one translation.

3.2 Usage of Minimal Rules

Despite the fact that the CSRS model can share information among instances using distributed word representations, it still poses an extremely sparse learning problem. Specifically, the numbers of unique subtrees $\tilde{t}$ and strings $\tilde{e}$ are extremely large,

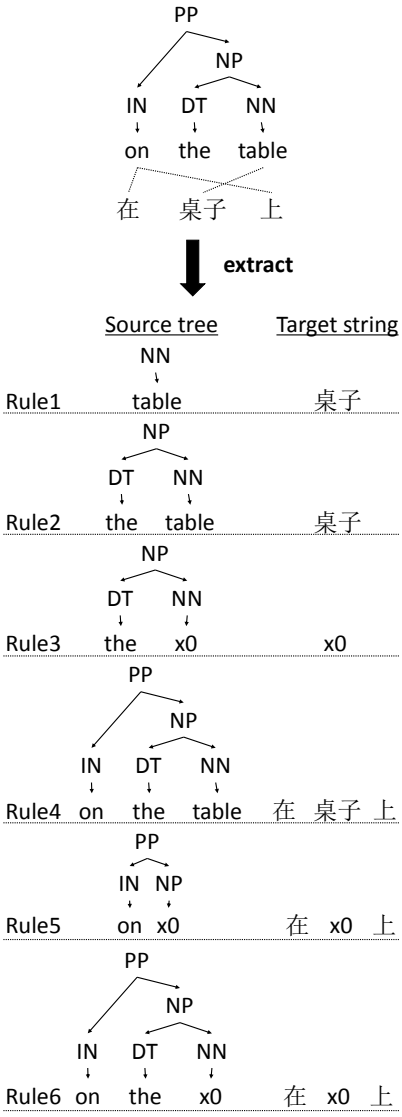

Figure 3: Rules.

and many may only appear a few times in the corpus. To reduce these problems of sparsity, we propose another improvement to the model, specifically through the use of minimal rules.

Minimal rules (Galley et al., 2004) are translation rules that cannot be split into two smaller rules. For example, in Figure 3, Rule2 is not a minimal rule, since Rule2 can be split into Rule1 and Rule3. In the same way, Rule4 and Rule6 are not minimal while Rule1, Rule3 and Rule5 are minimal.

Minimal rules are more frequent than non-minimal rules and have richer training data. Hence, we can expect that a rule selection model trained on minimal rules will suffer less from data sparsity problems. Besides, without non-minimal rules, the rule selection model will need less memory and can be trained faster.

To take advantage of this fact, we train another version of the CSRS model (CSRS-MINI) over only minimal rules. The probability of a non-minimal rule is then calculated using the product of the probability of minimal rules contained therein.

Note that for both the standard CSRS and CSRS-MINI models, we use the same baseline translation system which can use non-minimal translation rules. The CSRS-MINI model will break translation rules used in translations down into minimal rules and multiply all probabilities to calculate the necessary features.

4 Experiments

4.1 Setting

We evaluated the proposed approach for English-to-German (ED), English-to-French (EF), English-to-Chinese (EC) and English-to-Japanese (EJ) translation tasks. For the ED and EF tasks, the translation systems are trained on Europarl v7 parallel corpus and tested on the WMT 2015 translation task.[4] The test sets for the WMT 2014 translation task were used as development sets in our experiments. For the EC and EJ tasks, we used datasets provided for the patent machine translation task at NTCIR-9 (Goto et al., 2011).[5] The detailed statistics for training, development and test sets are given in Table 1. The word segmentation was done by BaseSeg (Zhao et al., 2006) for Chinese and Mecab[6] for Japanese.

For each translation task, we used Travatar (Neubig, 2013) to train a forest-to-string translation system. GIZA++ (Och and Ney, 2003) was used for word alignment. A 5-gram language model was trained on the target side of the training corpus using the IRST-LM Toolkit[7] with modified Kneser-Ney smoothing. Rule extraction was

[4]The WMT tasks provided other training corpora. We used only the Europarl corpus, because training a large-scale system on the whole data set requires large amounts of time and computational resources.

[5]Note that NTCIR-9 only contained a Chinese-to-English translation task. Because we want to test the proposed approach with a similarly accurate parsing model across our tasks, we used English as the source language in our experiments. In NTCIR-9, the development and test sets were both provided for the CE task while only the test set was provided for the EJ task. Therefore, we used the sentences from the NTCIR-8 EJ and JE test sets as the development set in our experiments.

[6]http://sourceforge.net/projects/mecab/files/

[7]http://hlt.fbk.eu/en/irstlm

			SOURCE	TARGET
ED	TRAIN	#Sents	1.90M	
		#Words	52.2M	49.7M
		#Vocab	113K	376K
	DEV	#Sents	3,003	
		#Words	67.6K	63.0K
	TEST	#Sents	2,169	
		#Words	46.8K	44.0K
EF	TRAIN	#Sents	1.99M	
		#Words	54.4M	60.4M
		#Vocab	114K	137K
	DEV	#Sents	3,003	
		#Words	71.1K	81.1K
	TEST	#Sents	1.5K	
		#Words	27.1K	29.8K
EC	TRAIN	#Sents	954K	
		#Words	40.4M	37.2M
		#Vocab	504K	288K
	DEV	#Sents	2K	
		#Words	77.5K	75.4K
	TEST	#Sents	2K	
		#Words	58.1K	55.5K
EJ	TRAIN	#Sents	3.14M	
		#Words	104M	118M
		#Vocab	273K	150K
	DEV	#Sents	2K	
		#Words	66.5K	74.6K
	TEST	#Sents	2K	
		#Words	70.6K	78.5K

Table 1: Data sets.

performed using the GHKM algorithm (Galley et al., 2006) and the maximum numbers of nonterminals and terminals contained in one rule were set to 2 and 10 respectively. Note that when extracting minimal rules, we release this limit. The decoding algorithm is the bottom-up forest-to-string decoding algorithm of Mi et al. (2008). For English parsing, we used Egret[8], which is able to output packed forests for decoding.

We trained the CSRS models (CSRS and CSRS-MINI) on translation rules extracted from the training set. Translation rules extracted from the development set were used as validation data for model training to avoid over-fitting. For different training epochs, we resample negative examples for each positive example to make use of different negative examples. The embedding dimension was set to be 50 and the number of hidden nodes was 100. The initial learning rate was set to be 0.1. The learning rate was halved each time the validation likelihood decreased. The number of epoches was set to be 20. A model was saved after each epoch and the model with highest validation likelihood was used in the translation system.

We implemented Liu et al. (2008)'s MERS model to compare with our approach. The training instances for their model were extracted from the training set. Following their work, the iteration number was set to be 100 and the Gaussian prior was set to be 1. We also compared the original MERS model and the MERS model trained only on minimal rules (MERS-MINI) to test the benefit of using minimal rules for model training.

[8]https://code.google.com/archive/p/egret-parser

	ED	EF	EC	EJ
Base	15.00	26.76	29.42	37.10
MERS	15.62	27.33	29.75	37.76
CSRS	16.15	28.05	30.12	37.83
MERS-MINI	15.77	28.13	30.53	38.14
CSRS-MINI	**16.49**	**28.30**	**31.63**	**38.32**

Table 2: Translation results. The bold numbers stand for the best systems.

	ED	EF	EC	EJ
CSRS vs. MERS	$>>$	$>>$	$>$	$-$
CSRS-MINI vs. MERS-MINI	$>>$	$-$	$>>$	$-$
MERS-MINI vs. MERS	$-$	$>>$	$>>$	$>>$
CSRS-MINI vs. CSRS	$>$	$-$	$>>$	$>>$

Table 3: Significance test results. The symbol $>>$ ($>$) represents a significant difference at the $p < 0.01$ ($p < 0.05$) level and the symbol - represents no significant difference at the $p < 0.05$ level.

The MERS and CSRS models were both used to calculate features used to rerank unique 1,000-best outputs of the baseline system. Tuning is performed to maximize BLEU score using minimum error rate training (Och, 2003).

4.2 Results

Table 2 shows the translation results and Table 3 shows significance test results using bootstrap resampling (Koehn, 2004): "Base" stands for the baseline system without any; "MERS", "CSRS", "MERS-MINI" and "CSRS-MINI" means the outputs of the baseline system were reranked using features from the MERS, CSRS, MERS-MINI and CSRS-MINI models respectively. Generally, the CSRS model outperformed the MERS model and the CSRS-MINI model outperformed the MERS-MINI model on different translation tasks. In addition, using minimal rules for model training benefitted both the MERS and CSRS models.

Table 4 shows translation examples in the EC task to demonstrate the reason why our approach improved accuracy. Among all translations, $T_{CSRS-MINI}$ is basically the same as the reference with only a few paraphrases that do not alter the meaning of the sentence. In con-

Source	typical dynamic response rate of an optical gap sensor as described above is approximately 2 khz , or 0.5 milliseconds .
Reference	上述(described above) 光学(optical) 间隙(gap) 传感器(sensor) 的 典型(typical) 动态(dynamic) 响应(response) 率(rate) 约(approximately) 为(is) 2KHz 或(or) 为 0.5 毫秒(milliseconds) 。
T_{Base}	典 型(typical) 的 动 态(dynamic) 响 应(response) 速 率(rate) 间隙(gap) 传感器(sensor) 的 光学(optical) 如上(above) 描述(described) 的 是(is) 约(approximately) 2 千赫(khz) 兹 ， 或(or) 0.5 毫秒(milliseconds) 。
T_{MERS}	典型(typical) 的 动态(dynamic) 响应(response) 率(rate) 光学(optical) 传感器(sensor) ， 如(as) 以上(above) 所 述(described) 间隙(gap) 的 约(approximately) 2 千赫(khz) ， 或(or) 0.5 毫秒(milliseconds) 。
T_{CSRS}	光学(optical) 传感器(sensor) ， 如(as) 以上(above) 所 述(described) 间隙(gap) 的 典型(typical) 的 动态(dynamic) 响应(response) 速率(rate) 为(is) 约(approximately) 2 千赫(khz) 兹 ， 或(or) 0.5 毫秒(milliseconds) 。
$T_{MERS-MINI}$	典 型(typical) 的 动 态(dynamic) 响 应(response) 率(rate) 间隙(gap) 传感器(sensor) 的 光学(optical) 如 上(above) 描 述(described) 的 是(is) 约(approximately) 2 千 赫(khz) 兹 ， 或(or) 0.5 毫秒(milliseconds) 。
$T_{CSRS-MINI}$	如 上(above) 描 述(described) 的 光学(optical) 间隙(gap) 传感器(sensor) 典 型(typical) 的 动 态(dynamic) 响应(response) 速率(rate) 为(is) 约(approximately) 2 千赫(khz) 兹 ， 或(or) 0.5 毫秒(milliseconds) 。

Table 4: Translation examples.

R_1: T_{MERS}&T_{CSRS}	PP (IN (“of”) NP (NP (DT (“an”) NP’ (JJ (“optical”) x0:NN)) x1:NP’)) → “光学(optical)” x1 x0 “的”
R_2: $T_{MERS-MINI}$	PP (IN (“of”) NP (NP (DT (“an”) NP’ (JJ (“optical”) x0:NP’)) x1:SBAR)) → x0 “的” “光学(optical)” x1 “的”
R_3 : $T_{CSRS-MINI}$	NP’ (JJ (“optical”) x0:NP’) → “光学(optical)” x0

Table 5: Rules used to translate the source word “optical” in different translations. Shadows (R_3) stand for ambiguous rules.

trast, T_{Base}, T_{MERS}, T_{CSRS} and $T_{MERS-MINI}$ all contain apparent mistakes. For example, the source phrase “optical gap sensor” (covered by gray shadows in Table 4) is wrongly translated in T_{Base}, T_{MERS}, T_{CSRS} and $T_{MERS-MINI}$ due to incorrect reorderings.

Table 5 shows rules used to translate the source word “optical” in different translations: R_1 is used in T_{MERS} and T_{CSRS}; R_2 is used in $T_{MERS-MINI}$; R_3 is used in $T_{CSRS-MINI}$. Although the source word “optical” is translated to the correct translation “光学(optical)” in all translations, R_1, R_2 and R_3 cause different reorderings for the source phrase “optical gap sensor”. R_3 reorders this source phrase correctly while R_1 and R_2 cause wrong reorderings for this source phrase.

We can see that R_1 is umambiguous, so the MERS and CSRS models will give probability 1 to R_1, which could make the MERS and CSRS models prefer T_{MERS} and T_{CSRS}. This is a typical translation error caused by sparse rules since the source subtree in R_1 does not have other translations in the training corpus.

To compare the MERS-MINI and CSRS-MINI models, Table 6 shows minimal rules (R_{2a}, R_{2b}, R_{3a} and R_{3b}) contained in R_2 and R_3. Table 7 shows probabilities of these minimal rules calculated by the MERS-MINI and CSRS-MINI models respectively. We can see that the CSRS-MINI model gave higher scores for the correct translation rules R_{3a} and R_{3b} than the MERS-MINI model, while the MERS-MINI model gave a higher score to the incorrect rule R_{2b} than the CSRS-MINI model.

Note that R_{2b} and R_{3b} are the same rule, but the target-side context in $T_{MERS-MINI}$ and $T_{CSRS-MINI}$ is different. The CSRS-MINI model will give R_{2b} and R_{3b} different scores because the CSRS-MINI model used target-side context. However, the MERS-MINI model only used source-side features and gave R_{2b} and R_{3b} the same score. The fact that the CSRS-MINI model

R_{2a}	PP (IN (“of”) NP (NP (DT (“an”) NP’ (x0:JJ x1:NP’)) x2:SBAR)) → x1 “的” x0 x2 “的”
R_{2b}	JJ (“optical”) → “光学(optical)”
R_{3a}	NP’ (x0:JJ x1:NP’) → x0 x1
R_{3b}	JJ (“optical”) → “光学(optical)”

Table 6: Minimal rules contained in R_2 and R_3. Shadows (R_{2b}, R_{3a} and R_{3b}) stand for ambiguous rules.

	MERS-MINI	CSRS-MINI
R_{2a}	1	1
R_{2b}	0.5441	0.09632
R_{3a}	0.9943	0.9987
R_{3b}	0.5441	0.7317

Table 7: Scores of minimal rules.

gave a higher score for R_{3b} than R_{2b} means that the CSRS-MINI model predicted the target string in R_{2b} and R_{3b} is a good translation in the context of $T_{CSRS-MINI}$ but not so good in the context of $T_{MERS-MINI}$. As we can see, the target phrase “如上(above) 描述(described) 的(of) 光学(optical) 间隙(gap) 传感器(sensor)” around “光学(optical)” in $T_{CSRS-MINI}$ is a reasonable Chinese phrase while the target phrase “间隙(gap) 传感器(sensor) 的(of) 光学(optical) 如上(above) 描述(described) 的(of)” around “光学(optical)” in $T_{MERS-MINI}$ does not make sense. Namely, the CSRS model trained with target-side context can perform rule selection considering target sentence fluency, which is the reason why target-side context can help in the rule selection task.

4.3 Analysis

To analyze the influence of different features, we trained the MERS model using source-side and target-side n-gram lexical features similar to the CSRS model. When using this feature set, the performance of the MERS model dropped significantly. This indicates that the syntactic, POS and span features used in the original MERS model are important for their model, since these features can generalize better. Purely lexical features are less effective due to sparsity problems when training one maximum entropy based classifier for each ambiguous source subtree and training data for each classifier is quite limited. In contrast, the CSRS model is trained in a continuous space and does not split training data, which relieves the sparsity problem of lexical features. As a result, the CSRS model achieved better performance using only lexical features compared to the MERS model. We also tried to use pre-trained word embedding features for the MERS model, but it did not improve the performance of the MERS model, which indicates that the log-linear model is not able to benefit from distributed representations as well as the neural network model.

We also tried reranking with both the CSRS and MERS models added as features, but it did not achieve further improvement compared to only using the CSRS model. This indicates that although these two models use different type of features, the information contained in these features are similar. For example, the POS features used in the MERS model and the distributed representations used in the CSRS model are both used for better generalization.

In addition, using both the CSRS and CSRS-MINI models did not improve over using only the CSRS-MINI model in our experiments. There are two main differences between the CSRS and CSRS-MINI models. First, minimal rules are more frequent and have more training data than non-minimal rules, which is why the CSRS-MINI model is more robust than the CSRS model. Second, non-minimal rules contain more information than minimal rules. For example, in Figure 3, Rule4 contains more information than Rule1, which could be an advantage for rule selection. However, the information contained in Rule4 will be considered as context features for Rule1. Therefore, this is no longer an advantage for the CSRS model as long as we use rich enough context features, which could be the reason why using both the CSRS and CSRS-MINI models cannot further improve the translation quality compared to using only the CSRS-MINI model.

5 Related Work

The rule selection problem for syntax-based SMT has received much attention. He et al. (2008) proposed a lexicalized rule selection model to perform context-sensitive rule selection for hierarchical phrase-base translation. Cui et al. (2010) introduced a joint rule selection model for hierarchical phrase-based translation, which also approximated the rule selection problem by a binary classification problem like our approach. However, these two models adopted linear classifiers similar to those used in the MERS model (Liu et al., 2008), which suffers more from the data sparsity

problem compared to the CSRS model.

There are also existing works that exploited neural networks to learn translation probabilities for translation rules used in the phrase-based translation model. Namely, these methods estimated translation probabilities for phrase pairs extracted from the parallel corpus. Schwenk (2012) proposed a continuous space translation model, which calculated the translation probability for each word in the target phrase and then multiplied the probabilities together as the translation probability of the phrase pair. Gao et al. (2014) and Zhang et al. (2014) proposed methods to learn continuous space phrase representations and use the similarity between the source and target phrases as translation probabilities for phrase pairs. All these three methods can only be used for the phrase-based translation model, not for syntax-based translation models.

There are also works that used minimal rules for modeling. Vaswani et al. (2011) proposed a rule Markov model using minimal rules for both training and decoding to achieve a slimmer model, a faster decoder and comparable performance with using non-minimal rules. Durrani et al. (2013) proposed a method to model with minimal translation units and decode with phrases for phrase-based SMT to improve translation performances. Both of these two methods do not use distributed representations as used in our model for better generalization.

In addition, neural machine translation (NMT) has shown promising results recently (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015a; Jean et al., 2015; Luong et al., 2015b). NMT uses a recurrent neural network to encode the whole source sentence and then produce the target words one by one. These models can be trained on parallel corpora and do not need word alignments to be learned in advance. There are also neural translation models that are trained on word-aligned parallel corpus (Devlin et al., 2014; Meng et al., 2015; Zhang et al., 2015; Setiawan et al., 2015), which use the alignment information to decide which parts of the source sentence are more important for predicting one particular target word. All these models are trained on plain source and target sentences without considering any syntactic information while our neural model learns rule selection for tree-based translation rules and makes use of the tree structure of natural language for better translation. There is also a new syntactic NMT model (Eriguchi et al., 2016), which extends the original sequence-to-sequence NMT model with the source-side phrase structure. Although this model takes source-side syntax into consideration, it still produces target words one by one as a sequence. In contrast, the tree-based translation rules used in our model can take advantage of the hierarchical structures of both source and target languages.

6 Conclusion

In this paper, we propose a CSRS model for syntax-based SMT, which is learned by a feed-forward neural network on a continuous space. Compared with the previous MERS model that used discrete representations of words as features, the CSRS model uses real-valued vector representations of words and can exploit similarity information between words for better generalization. In addition, we propose to use only minimal rules for rule selection to further relieve the data sparsity problem, since minimal rules are more frequent and have richer training data. In our experiments, the CSRS model outperformed the previous MERS model and the usage of minimal rules benefitted both CSRS and MERS models on different translation tasks.

For future work, we will explore more sophisticated features for the CSRS model, such as syntactic dependency relationships and head words, since only simple lexical features are used in the current incarnation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Lei Cui, Dongdong Zhang, Mu Li, Ming Zhou, and Tiejun Zhao. 2010. A joint rule selection model for hierarchical phrase-based translation. In *Proc. ACL*, pages 6–11.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proc. ACL*, pages 1370–1380.

Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013. Model with minimal translation units, but decode with phrases. In *Proc. HLT-NAACL*, pages 1–11.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. *arXiv preprint arXiv:1603.06075*.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proc. HLT-NAACL*, pages 273–280.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. COLING-ACL*, pages 961–968.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *Proc. ACL*, pages 699–709.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K Tsou. 2011. Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proc. NTCIR-9*, pages 559–578.

Jonathan Graehl and Kevin Knight. 2004. Training tree transducers. In *Proc. HLT-NAACL*, pages 105–112.

Zhongjun He, Qun Liu, and Shouxun Lin. 2008. Improving statistical machine translation using lexicalized rule selection. In *Proc. Coling*, pages 321–328.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proc. ACL-IJCNLP*, pages 1–10.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP*, pages 388–395.

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

Qun Liu, Zhongjun He, Yang Liu, and Shouxun Lin. 2008. Maximum entropy based rule selection model for syntax-based statistical machine translation. In *Proc. EMNLP*, pages 89–97.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*, pages 1412–1421.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015b. Addressing the rare word problem in neural machine translation. In *Proc. ACL-IJCNLP*, pages 11–19.

Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu. 2015. Encoding source language with convolutional neural network for machine translation. In *Proc. ACL-IJCNLP*, pages 20–30.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proc. HLT-ACL*, pages 192–199.

Graham Neubig. 2013. Travatar: A forest-to-string machine translation engine based on tree transducers. In *Proc. ACL*, pages 91–96.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

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

Holger Schwenk. 2012. Continuous space translation models for phrase-based statistical machine translation. In *Proc. COLING*, pages 1071–1080.

Hendra Setiawan, Zhongqiang Huang, Jacob Devlin, Thomas Lamar, Rabih Zbib, Richard Schwartz, and John Makhoul. 2015. Statistical machine translation features with multitask tensor networks. In *Proc. ACL-IJCNLP*, pages 31–41.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Haitao Mi, Liang Huang, and David Chiang. 2011. Rule markov models for fast tree-to-string translation. In *Proc. HLT-ACL*, pages 856–864.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proc. EMNLP*, pages 1387–1392.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proc. ACL*, pages 111–121.

Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, Graham Neubig, and Satoshi Nakamura. 2015. A binarized neural network joint model for machine translation. In *Proc. EMNLP*, pages 2094–2099.

Hai Zhao, Chang-Ning Huang, and Mu Li. 2006. An improved chinese word segmentation system with conditional random field. In *Proc. SIGHAN*, pages 162–165.