

Research Article

Filtering Reordering Table Using a Novel Recursive Autoencoder Model for Statistical Machine Translation

Jinying Kong,^{1,2,3} Yating Yang,^{1,2} Lei Wang,^{1,2} Xi Zhou,^{1,2} Tonghai Jiang,^{1,2} and Xiao Li^{1,2}

¹Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi 830011, China

²Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China

³University of Chinese Academy of Sciences, Beijing 100049, China

Correspondence should be addressed to Yating Yang; yangyt@ms.xjb.ac.cn

Received 6 March 2017; Revised 27 April 2017; Accepted 8 May 2017; Published 13 June 2017

Academic Editor: Zhongwei Lin

Copyright © 2017 Jinying Kong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In phrase-based machine translation (PBMT) systems, the reordering table and phrase table are very large and redundant. Unlike most previous works which aim to filter phrase table, this paper proposes a novel deep neural network model to prune reordering table. We cast the task as a deep learning problem where we jointly train two models: a generative model to implement rule embedding and a discriminative model to classify rules. The main contribution of this paper is that we optimize the reordering model in PBMT by filtering reordering table using a recursive autoencoder model. To evaluate the performance of the proposed model, we performed it on public corpus to measure its reordering ability. The experimental results show that our approach obtains high improvement in BLEU score with less scale of reordering table on two language pairs: English-Chinese (+0.28) and Uyghur-Chinese (+0.33) MT.

1. Introduction

Recently, machine learning model based on deep neural network (DNN) has achieved great breakthrough in many application fields. Furthermore, it is currently becoming a dominant method in both image recognition and automatic speech recognition [1]. Some DNN techniques such as autoencoder, long short-time memory (LSTM), and convolution neural network (CNN) have obtained satisfying results in the field of natural language processing (NLP) [2–4]. However, to the best of our knowledge, the idea of DNN has not achieved comparable success in NLP. This is due to the fact that, unlike image or voice, structure of the language is more complex and feature extraction is more difficult.

As a part of NLP, application of deep learning on machine translation (MT) can be divided into two types: Neural Machine Translation (NMT) and deep learning applied on PBMT [5, 6]. PBMT which is also called traditional machine translation now is facing the impact of NMT, which is a new neural-network-based model of MT. With the good translation performance and simple structure, NMT draws most attentions on application of neural network on the MT.

Despite these, NMT relies on huge size of corpus, and the argument about which has better performance between NMT and PBMT still continues. Furthermore, large size of corpus is impossible on some language pairs such as Uyghur-Chinese. In this study, we present a novel DNN model to optimize reordering model on PBMT.

PBMT generally extracts the phrase and reordering examples from the result of word alignment and then generates the phrase table and reordering table which would be used to the decoding process. The former can be called translation model and the latter can be called reordering model. Adding a language model [7], an integrated PBMT system can translate the input sentence by decoding. It is valuable to research language model, because it is not only used to machine translation but also applied to the other fields of NLP. Similarly, as a component of machine translation, the translation model is always a research focus. From the original word-based translation model to the latest NMT model, the performance of MT is getting better and better and MT is getting closer to automatic control [8, 9]. Unlike most works which aim to filter phrase table, this paper focuses on optimizing the reordering model in PBMT by filtering reordering

asian countries and	亚洲 乃至	other mono
asian countries and	亚洲 各国	other mono
asian countries and	亚洲 国家	other mono
asian countries and	亚洲 国家 和	mono mono

FIGURE 1: A part of reordering table in Moses system.

table. The previous reordering models are useful, except in environments where memory and time constraints are imposed. The proposed model can get better performance with less space.

In this paper, we propose a DNN model consisting of three parts: a generative model, a discriminative model, and a filtering strategy. The generative model is a recursive autoencoder to implement reordering rule embedding (compact vector representations for reordering rule). The discriminative model is a multilayer perceptron (MLP) to classify these rule vectors. The filtering strategy based on minimum difference is designed to filter reordering rule. Our model is used to reconsider the reordering table and filter its wrong and noisy rules. After this filtering process, the modified reordering table can successfully accelerate the speed of decoding and eventually improve the quality of translation. Figure 1 is a reordering table in Moses system from English-Chinese MT.

This paper is organized as follows. Section 1 is introduction, mainly introducing the background of our research. Section 2 is illustrations of some representative workings on reordering model. The details of our filtering model based on DNN are elaborated in Section 3. The settings and results of experiments on this DNN model are given in Section 4. Section 5 is conclusions and future works.

2. Related Work

There are many various methods that have been proposed to filter the phrase translation table on Statistical Machine Translation systems. Yin et al. proposed a method to filter the phrase table based on virtual context [10]. Di et al. proposed C-value and phrase sticky degree in this field [11]. Zens et al. use the basic principles of acoustics to filter phrase table [12]. Zhang et al. take advantage of a bilingually constrained recursive autoencoder to learn semantic phrase embedding and prune phrase table through phrase semantic similarities [13]. Compared with translation model, the reordering model is more independent, while few researchers presented related methods to filter the reordering table.

In Statistical Machine Translation systems, the reordering models are various from simple distance penalty model to complex machine learning models. The first type is following the simple principles; researchers believe that the language model and translation model are enough to accomplish the task of reordering. The representative work of this type is a simple distance penalty model proposed by Koehn et al. [14]. This model is simple to implement but effective in English-French MT. The second type is the current mainstream method; it has complex definitions of reordering orientations and discrimination of reordering orientations. The methods of predicting reordering orientations are various from simple

maximum likelihood to complex maximum entropy model [15, 16]. Li et al. proposed a DNN reordering model to discriminate reordering orientations [17]. The third type of reordering model uses information on syntax or grammar among different language pairs. This type of methods is generally used in the process of decoding and takes advantage of grammar rules to limit the words order of translation results. It seems like the rule-based machine translation model. For example, both of Xiao et al. and Wang et al. used the syntactic information of Chinese language to direct reordering operations [18, 19].

This paper presents a reordering table filtering model to improve the reordering ability of MT; our method optimizes reordering model belonging to the second type which is the most popular method for researchers [15–17]. This type of reordering model includes two factors: reordering orientations and the score of reordering orientations. The reordering orientations refer to the discrimination of join orders in two consecutive bilingual phrase pairs. The common reordering orientations are *monotone*, *swap*, and *discontinuous*. The discontinuous reordering orientation also can be divided into *discontinuous monotone* and *discontinuous swap*. Figure 2 is an example of reordering orientations with respect to the adjacent phrases. For example, the word “minister” is monotonous to its prior word. Formula (1) is the definition of four orientations: *monotone*, *swap*, *discontinuous monotone*, and *discontinuous swap*:

$$O_i = \begin{cases} \text{monotone,} & a_{b^2.r_f} - a_{b^1.r_i} = 1 \\ \text{swap,} & a_{b^2.r_i} - a_{b^1.r_f} = -1 \\ \text{discontinuous monotone,} & a_{b^2.r_f} - a_{b^1.r_i} > 1 \\ \text{discontinuous swap,} & a_{b^2.r_i} - a_{b^1.r_f} < -1, \end{cases} \quad (1)$$

where O denotes orientation type, b denotes the block (word or phrase), and a denotes the location in original sentence. The monotone orientation is an example that the location of the first word in second block follows the location of the last word in first block; other orientations follow the same principle. The way to score reordering orientations is a problem on pattern recognition, which can be solved by a simple method: accumulate the number of same rules and count the frequency of their occurrence. Meanwhile, we can take advantage of the machine learning model such as Naive Bayes model, maximum entropy model, or deep neural network model to discriminate reordering orientation in recent years.

All in all, the second type of reordering model treats reordering as a machine learning problem. A good machine learning system needs two points: training data with good quality and machine learning model of high performance. Unlike most of the previous reordering models that focus on machine learning model optimization, our methods optimize the training data. The previous works are useful, except in environments where memory and time constraints are imposed. Because of independence, our reordering table filtering model based on DNN can efficiently adapt to different reordering models. This paper optimizes the training set of machine learning models (reordering model) by DNN, so the

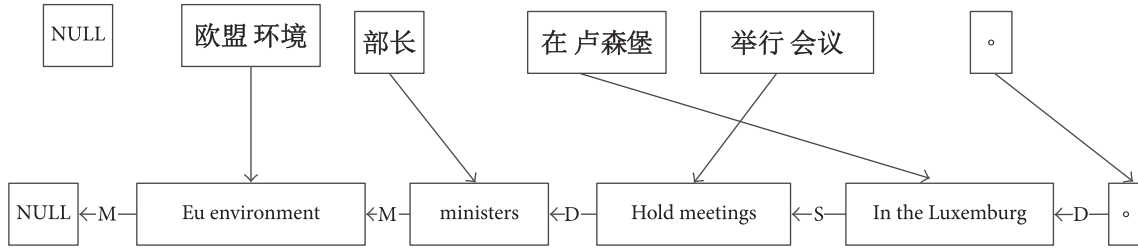


FIGURE 2: An example of reordering orientations with respect to the adjacent phrases.

time consumption of decoding and the quality of translation in MT is improved.

3. Reordering Table Filtering Model Based on DNN

The work flow of our model is shown as Figure 3. Firstly, we preprocess the reordering table to get an adaptable dataset. Secondly, we use generative model called recursive autoencoder to generate a continuous space representation which treats a rule as a dense real-valued vector. Thirdly, another discriminative model called multilayer perceptron (MLP) is used to score the orientation of each rule. Finally, according to the orientation score of each rule, we select the final reordering rule through the strategy of minimum difference. We use the stochastic gradient descent to adjust the parameters of whole model; stochastic gradient descent is a good way to adjust parameters [20–22]. Here, we first introduce the text preprocessing of original reordering table and then describe the construction of autoencoder-based generative model and MLP-based discriminative model as well as the filtering strategy based on minimum difference.

3.1. Text Preprocessing. Figure 1 is an example that shows the reordering table in Moses systems. These rules are extracted from various corpus and vary in quality. It is a large and redundancy dataset. Furthermore, because computation consumption of DNN is enormous, too large training set is easy to cause too much computations and hard to achieve convergence. Therefore, we need to preprocess the reordering table before training our model. In view of various reordering rules from different models, we summarize these general characteristics of reordering table:

- (1) Same rules account for about 30 percent of the total number.
- (2) There are many short rules, most of them can merge to their corresponding long rules, the rule whose length below five is about eighty percent of the total number (the maximum rule length is 7).
- (3) There are many noise and useless data.
- (4) More than 90 percent of rules with only one word are ambiguous of reordering orientations.

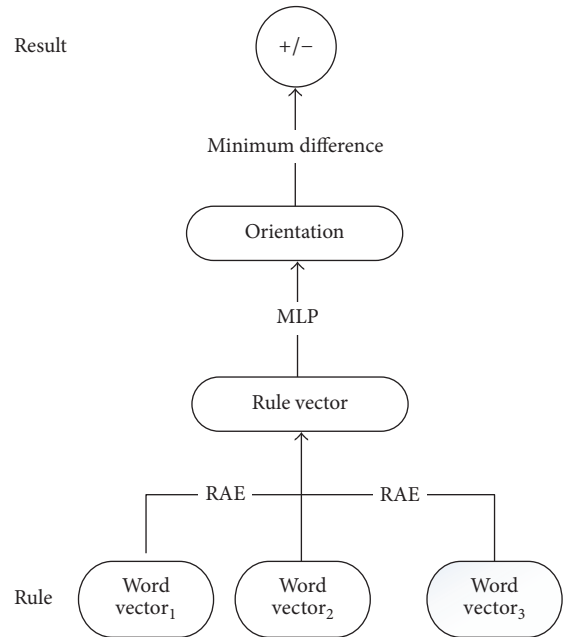


FIGURE 3: The work flow of DNN-based reordering table filtering model.

According to the descriptions from 1 to 4, this paper deals with the reordering table as follows:

- (1) Adding an attribute to every reordering rule in order to record the number of this rule, especially for rules with one word
- (2) Deleting redundancy rules and only saving one as well as recording the total number of this reordering rule
- (3) In order to accelerate training, combining some short rules to long rule in situation as shown in Table 1, then accumulating their number

For example, if rule 1 is “A, B, M, M” and rule 2 is “XA, YB, M, M,” which means the phrase pair “A, B” is monotonous with their prior and following phrases, and rule 2 reveals this situation, we combine them together. Table 1 shows the rules in reordering table which can be merged. In Table 1, “O₁” determine orientation with respect to prior phrase and “O₂” determine orientation with respect to following phrase. “TIPS” denotes which special orientations should appear.

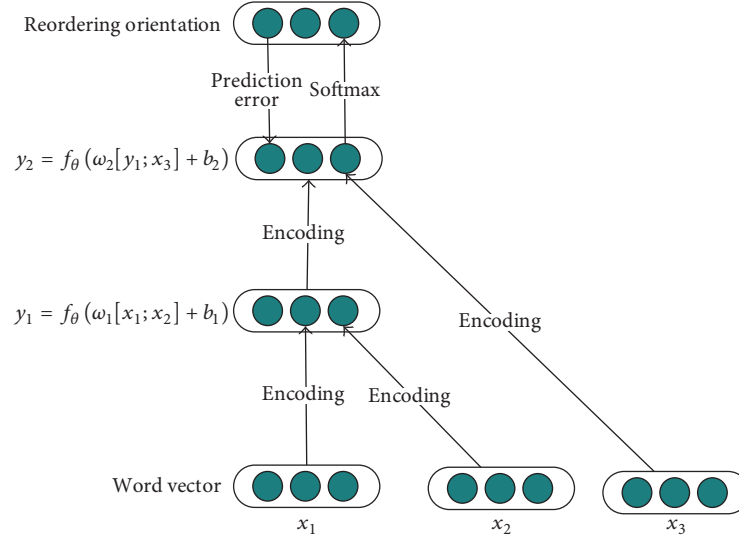


FIGURE 4: An illustration of the generative model based on ARE.

TABLE 1: The rules in reordering table can be merged.

RULE ₁	RULE ₂	TIPs
A, B, O ₁ , O ₂	XA, YB, O ₃ , O ₂	O ₁ = M
A, B, O ₁ , O ₂	AX, BY, O ₁ , O ₃	O ₂ = M
A, B, O ₁ , O ₂	XAW, WBV, O ₃ , O ₄	O ₁ = M and O ₂ = M

This paper focuses on how to filter reordering rule with wrong reordering orientations. The details of DNN-based classifier which is trained to reconsider the reordering score table are described in the following sections. The aim of our model is to select high-quality rules to retrain the reordering model and improve the quality of translation.

3.2. The Classified Model of Reordering Rules Based on DNN. The reordering table filtering algorithm consists of two components: a generative model and a discriminative model. In generative model, we use RAE (recursive autoencoder) to embed reordering rule. In discriminative model, MLP is used to score the orientation of each of the rules. RAE provides a reasonable composition mechanism to embed each rule and MLP is a simple but effective classifier based on deep learning [23, 24].

It is a classification issue that scores orientations of rules in reordering table. For example, if a reordering model has two reordering orientations such as *monotone* and *swap*, there are four types in reordering table: “*swap, monotone*,” “*monotone, swap*,” “*swap, swap*,” and “*monotone, monotone*.” In addition, the length of rules in reordering table generally is less than ten, so filtering reordering table can be seen as a classification problem of short texts. The problem of short text classification is not a trivial one. Since the feature vectors of text are always high dimension and sparseness, the result of short text classification is far from satisfactory.

Autoencoder can simulate human brain and combine high dimension of features in a nonlinear way to obtain the

low dimension of abstract features [25–27]; it is an advanced model of machine learning. MLP accepts input vectors and can easily enhance classifying performance by adding hide layers [28].

3.2.1. Word Embedding. The reordering rules consists of source phrase, target phrases, and the reordering orientations. In rule embedding process, the word vector is the basis and serves as the input to the generative model. After learning word embedding, all vectors are stacked into an embedding matrix $L \in R^{n \times |V|}$, and each word in our vocabulary V corresponds to a vector $x \in R^n$.

Given a reordering rule which is an ordered list of m words, each word has a column index i of the embedding matrix L . The index i is used to retrieve the word’s vector representation using a simple multiplication with a one-hot vector e which is zero in all positions except for the i th index:

$$x_i = Le_i \in R^n. \quad (2)$$

According to previous researches [13], n is usually set empirically, such as $n = 50, 100, 200$.

3.2.2. Generative Model. Generator is a semisupervised rule embedding model which can learn vector representation and can be well adapted to the given label. Assuming we are given a reordering rule, it is first projected into a list vectors $(x_1, x_2, x_3, \dots, x_m)$ by using formulation(2). The RAE learns the rule vector representation by recursively combining two children word vectors in a bottom-up manner. As shown in Figure 4, the recursive autoencoder accepts input data and works. The details are as follows:

- (1) Putting a nonlinear change on the input vector, we choose an element-wise activation function such as $f = \tan h(\cdot)$ and obtain the encoding result y through

it. This step is called encoding, as shown in formula (3). We should notice that x at here means $[c1; c2] \in R^{2n \times 1}$ and matrix w here means $w \in R^{n \times 2n}$, so that wx is still a vector with the same dimension as input vector and so does y :

$$y = f_{\theta}(wx + b). \quad (3)$$

- (2) The encoding result y is restructured by decoder and outputs its corresponding vector z . W^T is transposition of W ; both b and b' are offset. This step is called decoding, as shown in the following formula:

$$z = g_{\theta'}(y) = a(W^T y + b'). \quad (4)$$

- (3) We use regularization cost function to estimate the similarity between x and z . This step is called estimation, as shown in the following formula:

$$L(x, z) = KL(x \| z) + a \sum_{i=0}^{|\theta|} |\theta_j|. \quad (5)$$

- (4) We use stochastic gradient descent algorithm to update the parameters to minimize the cost function between x and z . This step is called optimization, as shown in the following formulas:

$$\begin{aligned} W &\leftarrow W - l \times \frac{\partial L(x, z)}{\partial W} \\ \theta^* &= \arg \min \{L(x, z)\} \\ &= \arg \min \{Lx, g_{\theta'}(f_{\theta}(\tilde{x}))\}. \end{aligned} \quad (6)$$

- (5) y which extracted from above four steps is the rule feature vector, and then add some noise to y as input vector to encoding. The deep recursive autoencoder is obtained by using above four steps for many circulations. To avoid too much calculations, the number of iteration is set to 50–80.

The deep network has good characteristics to abstraction and feature extraction. The above RAE is completely unsupervised and can only induce general representation of the multiword. Here we add a softmax layer to extend the original RAEs to a semisupervised model. At the last layer, the objective function includes the reconstruction error and the prediction error, as shown in the following formula:

$$E(X, T; \theta) = \alpha E_{\text{rec}}(x, t; \theta) + (1 - \alpha) E_{\text{pred}}(x, t; \theta). \quad (7)$$

3.2.3. Discriminative Model. After obtaining abstract vectors from RAE, we use MLP as classifier to implement reordering rule classification. Like common MLP, our discriminative model consists of input layer, hide layers, and output layer. Input layer accepts abstract vectors from RAE, hides layers consisting of neurons whose activation function is $f = \tan h()$, and extracts features from input, and output layer is a softmax layer. The number of units in output layer is equal to the number of reordering orientations. Softmax regression

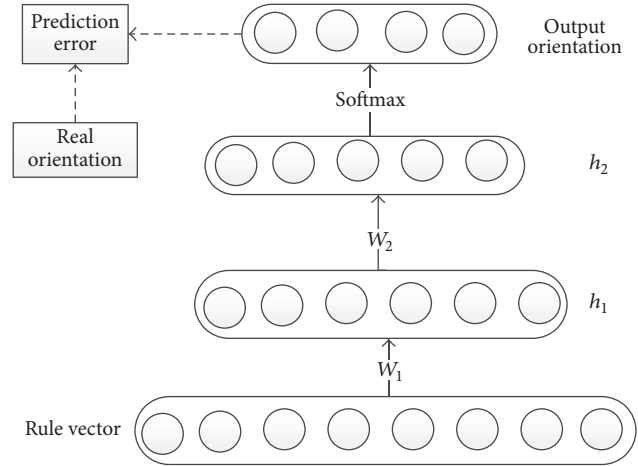


FIGURE 5: A flow diagram of the reordering discriminative model based on MLP.

is a multiple classification of logistic regression; formula (8) shows the definition of softmax function:

$$P(y = o | x) = \text{softmax}(w_o^T x) = \frac{w_o^T x}{\exp(\sum_{i=1}^o w_i^T x)}. \quad (8)$$

Every component of the output is a score corresponding to the reordering orientation probability according to input rule. After adding softmax layer, we use the pretrained weights as initial weights and minimize supervised cost between the output probability and real reordering orientation probability to modify overall parameters of the network. Figure 5 is a flow diagram of the reordering classification model based on MLP.

3.3. Filtering Strategy. After above steps, we can obtain a DNN-based classifier which accurately outputs each reordering orientation score of reordering rules. This paper defines a standard estimation to evaluate the quality of reordering rules; it is represented by the following formula:

$$\text{accuracy}(y_i) = [(\max(\text{score}_i)) - \text{score}_i(O_s)], \quad (9)$$

where $\max(\text{score}_i)$ refers to the maximal distributed score of the DNN-based classifier and also the most reasonable reordering orientation of the reordering rule. $\text{score}_i(O_s)$ refers to the probability of original reordering orientation in the DNN-based classifier. In other words, the accuracy of reordering orientation refers to the different value of reordering orientation probability between original reordering orientation and the most reasonable reordering orientation in classifier. When this value is equal to zero, it indicates that this reordering rule is positive because the reordering orientation in original reordering table is the same with the most reasonable reordering orientation in classifier.

For example, a reordering orientation in original reordering table is “*monotone, monotone,*” and the most reasonable reordering orientation in classifier is also “*monotone, monotone,*” so the accuracy of reordering orientation is zero and this rule is positive.

TABLE 2: The corpus of our experiments.

Category	Training set	Development set	Test set
Uyghur-Chinese	139,792	1,100	1,000
English-Chinese	7780,000	1,000	1,100

Filtering strategy based on minimum difference refers to use DNN-based classifier to calculate the accuracy of every rule in reordering table and then rerank reordering rule by accuracy score in ascending order. Finally, we choose a scale of reordering table according to the quality of original training corpus. In general case, we select sixty percent of the original reordering table whose performance is comparative with original reordering table.

4. Experiments

We applied the proposed model to phrase-based machine translation systems to evaluate its performance. Our experiments include English-Chinese and Uyghur-Chinese translation.

4.1. Settings. The corpus come from the CWMT 2015 public evaluation datasets and we use English-Chinese and Uyghur-Chinese corpus in news domain as our research objects. Since our model is used to machine translation decoding, we divided the corpus into three parts: training set, test set, and development set. The details of corpus are shown in Table 2. The parallel English-Chinese training data from CWMT contains 77.8 M sentence pair. The parallel Uyghur-Chinese training data from CWMT contains about 0.14 M sentence pair. The development set of English-Chinese contains 1 K sentences. The development set of Uyghur-Chinese contains 1.1 K sentences. Both test sets of Uyghur-Chinese and English-Chinese contain 1 k sentences.

We firstly pretrain the word embedding with toolkit Gensim (<https://is.muni.cz/publication/884893/en>) on training set. For the dimensionality, we set it as 50. Then the reordering rule representation is learned by a RAE model shared by Socher (<https://github.com/jeremysalwen/ParaphraseAuto-encoder-octave>) in GitHub. We empirically set the learning rate as 0.01. The discriminative model that is chosen by us is implemented by Theano (<http://deeplearning.net/tutorial/mlp.html#mlp>). The MLP is a deep learning network including four layers; the number of neurons in each layer, respectively, is 50, 80, 80, and n. The number of neurons in input layer is decided by the dimension of word embedding; the number of neurons in output layer is decided by the types of reordering orientation. Besides, we set learning rate as 0.1 and weight penalty factor as 0.0002 in stochastic gradient descent algorithm.

The proposed method was executed on a computer with Moses 2.1 (<http://www.statmt.org/moses/>), 4 GB memory, and Ubuntu 12.04. The word alignment tool which we selected is open source GIZA++ and then we use the strategy of “grow-diag-final-and” to implement many-to-many word alignments. The maximal extracted phrase length is 7 and the reordering model is selected as the variate in various

experiments. In process of tuning parameters, we use MERT method to optimize arguments. In addition, we use SRILM to training two 5-gram language models on each Chinese corpus and estimate parameters according to Kneser-Ney smoothing algorithm. The evaluation metric of machine translation is case-insensitive BLEU-4 scores [29].

In order to compare with various reordering methods, we take two experiment tasks of English-Chinese and Uyghur-Chinese into consideration. The settings of both experiments are the same and we also set five small groups in these two experiments; the details are as follows.

Baseline. We use default distance penalty model as our reordering model to train translation model; this test has no reordering table.

MSD. We use the option “*phrase-msd-bidirectional-fe*” as reordering model to train translation model; “*phrase*” denotes this is a phrase-based MT model; “*msd*” denotes that this model has three orientations: *monotone*, *swap*, and *discontinuous*; “*bidirectional*” denotes that this model determines orientation with respect to both following and prior phrase; “*fe*” denotes that this model conditions on both the source and target languages.

MSD.F. We firstly use the option “*phrase-msd-bidirectional-f*” as reordering model to train translation model, and then utilize our proposed filtering model to select 80, 60 and 40 percent size of original reordering table respectively. Finally, the filtered reordering table are used to retrain the reordering model which would be used in decoding.

MSLR. We use the option “*phrase-mslr-bidirectional-fe*” as reordering model to train translation model. All parameters are the same means with above. Besides, “*mslr*” denotes that this reordering model has four orientations: *monotone*, *swap*, *discontinuous-left*, and *discontinuous-right*.

MSLR.F. We use the option “*phrase-mslr-bidirectional-f*” as reordering model to train translation model and then utilize our proposed filtering model to select 80, 60, and 40 percent size of original reordering table, respectively. Finally, the filtered reordering table is used to retrain the reordering model which would be used in decoding.

4.2. Results. Tables 3 and 4 demonstrated the experiment results of English-Chinese and Uyghur-Chinese machine translation system, respectively. Figures 6 and 7 showed the currency of average BLEU score on Uyghur-Chinese and English-Chinese MT, respectively. In Figures 6 and 7, the numbers 1–9 denote the 9 small groups in our experiments. According to the results of experiments, we can draw following conclusions.

The performance of machine translation systems gets improvement by applying the reordering table filtering model based on DNN. Compared with the original reordering model, the pruned one needs less space but is more useful. The BLEU score gains 0.15 improvement on average while the size of reordering table is 80 percent of original reordering

TABLE 3: Experiment results of Uyghur-Chinese MT.

Group	Size of reordering table	Development set	Test set	Average
Baseline	N/A	35.72	35.14	35.43
MSD	645 MB	37.05	36.34	36.70
	516 MB	37.20	36.58	36.89 (+0.19)
	387 MB	37.22	36.70	36.96 (+0.26)
MSD_F	387 MB	36.83	36.23	36.53 (-0.17)
	756 MB	37.12	36.44	36.78
MSLR	605 MB	37.25	36.68	36.97 (+0.19)
	454 MB	37.45	36.77	37.11 (+0.33)
	302 MB	37.01	36.23	36.62 (-0.16)

TABLE 4: Experiment results of English-Chinese MT.

Group	Size of reordering table	Development set	Test set	Average
Baseline	N/A	29.45	30.12	29.79
MSD	5084 MB	30.13	30.35	30.24
	4067 MB	30.25	30.41	30.33 (+0.9)
	3050 MB	30.30	30.48	30.39 (+0.15)
MSD_F	2033 MB	29.80	30.20	30.00 (-0.33)
	5763 MB	30.15	30.35	30.25
MSLR	4610 MB	30.31	30.42	30.37 (+0.12)
	3457 MB	30.55	30.51	30.53 (+0.28)
	2305 MB	30.06	30.21	30.14 (-0.23)

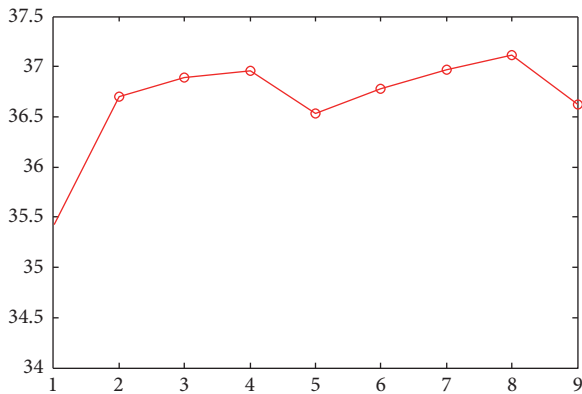


FIGURE 6: The currency of average BLEU score on Uyghur-Chinese MT.

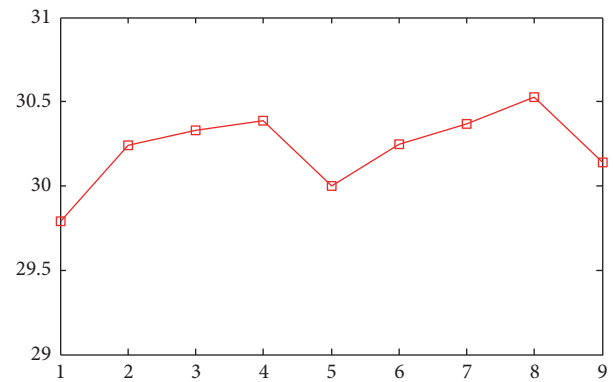


FIGURE 7: The currency of average BLEU score on English-Chinese MT.

table; the BLEU score can gain 0.23 on average improvement while the size of reordering table is just 60 percent of original reordering table. However, the BLEU score reduces 0.22 on average when the size of reordering table is 40 percent of original reordering table. In addition, the best performance of Uyghur-Chinese machine translation system obtains the improvement of 0.33 BLEU score, and English-Chinese MT is 0.28.

Influence of our model on machine translation varies on different language pair. In experiment, the improvement of English-Chinese machine translation is not so obvious compared with Uyghur-Chinese. As we can see from the result, the former gets less improvement than the latter. The reason

is probably that grammatical differences between Uyghur and Chinese are more than English and Chinese, so the reordering problems in Uyghur-Chinese MT are more prominent than English-Chinese MT. Therefore, the performance of our model is better for MT with more grammatical differences.

Our model gets more improvements with more types of reordering orientations. We can see from Table 3 that reordering model with four orientations gains 0.26 improvement on average, and the reordering model with three orientations gains 0.225; this situation happened again in Table 4. The reason may be that orientations discrimination by reordering model with four orientations is more dependent on the quality of the training set than the reordering model with three

orientations. And filtering the reordering table by our DNN model helps enhance the accuracy of classifier which discriminate the orientation when decoding.

Besides, our model is also influenced by the correlation between training set and test set. We found that the average BLEU score in development set is higher than test set in English-Chinese MT; this situation is opposite to Uyghur-Chinese MT, which means the training data has more correlations with test set than development set in English-Chinese MT. On the contrary, the BLEU score gains 0.19 improvement on development set and 0.16 improvement on test set on average in English-Chinese MT. The same situation happens in Uyghur-Chinese MT whose development set has more correlations with training data and gains less improvement than test set. As deep neural network is more powerful on filtering noise data than traditional machine learning methods, it means our model prefers dirty data than clean data. On the other words, our model is more suitable for MT whose test data has less correlation with training data.

Finally, we found that our model can achieve best performance when the size of filtered reordering table is 60 percent of original reordering table. The reason is that the selected reordering rules can cover original reordering table and obtain more accurate probability of reordering orientations with this proportion. When the size of original reordering table was reduced to 40 percent, some reordering knowledge has been dropped. While the size of original reordering table retains 80 percent, the improvements are not so obvious because the difference between the original reordering table and filtered reordering table is too little.

All in all, this model is suitable for machine translation systems based on arbitrary language pair when the machine translation generates reordering table in the process of training. Our model can improve the quality of machine translation in the situation of reducing the scale of reordering table and speed up the decoding process.

5. Conclusion

This paper proposed a reordering table filtering model based on deep neural network to improve the problems of reordering in Statistical Machine Translation. The proposed model is evaluated on the field of Uyghur-Chinese and English-Chinese machine translation. The experiment results show that the quality of machine translation in Uyghur-Chinese and English-Chinese obtains obvious improvements when using the new filtered reordering table in decoding process and the reordering ability gets improved.

To enhance the speed and accuracy of decoding in SMT, we optimize the reordering model by pruning reordering table. Reordering table consists of reordering rule and its corresponding orientation. Our method firstly filters the original reordering table by DNN-based model and then uses the filtered reordering rule to retrain the reordering model.

The paper focuses on reordering table, so the method we proposed can be used in any machine translation systems generating reordering table. However, not all machine translation systems generate reordering table, such as the translation model based on syntax. Meanwhile, our model is

independent of reordering model and the ability of reordering relies on the performance of reordering model. In future work, we plan to merge the reordering model based on DNN to PBMT as a feature function.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

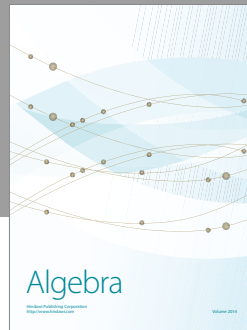
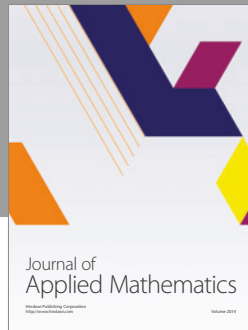
Acknowledgments

This work is supported by the Xinjiang Key Laboratory Fund under Grant no. 2015KL031, the Natural Science Foundation of Xinjiang (2015211B034), Xinjiang Science and Technology Major Project (2016A03007-3), and the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant no. XDA06030400.

References

- [1] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.
- [2] L. O. Chua and T. Roska, "The CNN paradigm," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 3, pp. 147–156, 1993.
- [3] Y. Bengio, H. Schwenk, J. Senécal et al., "Neural probabilistic language models," *Innovations in Machine Learning*, vol. 2006, pp. 137–186, 2006.
- [4] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
- [5] P. Koehn, R. Zens, C. Dyer et al., "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (ACL '07)*, pp. 177–180, Prague, Czech Republic, June 2007.
- [6] P. Brown, V. Pietra, S. Pietra et al., "The mathematics of statistical machine translation: parameter estimation," *Computational linguistics*, vol. 19, no. 2, pp. 263–311, 1993.
- [7] A. Stolcke, J. Zheng, W. Wang et al., "SRILM at sixteen: update and outlook," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, vol. 2011, pp. 5–6, 2011.
- [8] Z. Lin, J. Liu, W. Zhang, and Y. Niu, "Stabilization of interconnected nonlinear stochastic Markovian jump systems via dissipativity approach," *Automatica*, vol. 47, no. 12, pp. 2796–2800, 2011.
- [9] Z. Lin, W. Zhang, and Y. Lin, "Suboptimal stochastic H-two/H_∞ design with spectrum constraint," *Journal of Control Theory and Applications*, vol. 6, no. 3, pp. 317–321, 2008.
- [10] L. Yin, Y. Zhang, and J. Xu, "Phrase table filtration based on Virtual Context in Phrase-Based statistical Machine Translation," *Journal of Chinese Information Processing*, vol. 27, no. 6, pp. 139–143, 2013.
- [11] P. Di, Y. Zhou, and Z. Gong, "Phrase table filtration in Phrase-based Statistical Machine Translation," *Computer Applications and Software*, vol. 28, no. 5, pp. 28–30, 2011.

- [12] R. Zens, D. Stanton, and P. Xu, "A systematic comparison of phrase table pruning techniques," in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12)*, pp. 972–983, July 2012.
- [13] J. Zhang, S. Liu, M. Li, M. Zhou, and C. Zong, "Bilingually-constrained Phrase Embeddings for Machine Translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 111–121, Baltimore, Maryland, June 2014.
- [14] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 48–54, Edmonton, Canada, May 2003.
- [15] C. Tillmann and T. Zhang, "A localized prediction model for statistical machine translation," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pp. 557–564, June 2005.
- [16] D. Xiong, Q. Liu, and S. Lin, "Maximum entropy based phrase reordering model for statistical machine translation," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL '06)*, pp. 521–528, Sydney, Australia, July 2006.
- [17] P. Li, Y. Liu, and M. Sun, "Recursive autoencoders for ITG-based translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '13)*, pp. 567–577, October 2013.
- [18] X. Xiao, Y. Liu, and Q. Liu, "Lexical reordering for hierarchical phrased-based translation," *Journal of Chinese Information Processing*, vol. 26, no. 1, pp. 37–41, 2006.
- [19] C. Wang, M. Collins, and P. Koehn, "Chinese syntactic reordering for statistical machine translation," in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '07)*, pp. 737–745, June 2007.
- [20] W. Zhang and B.-S. Chen, "On stabilizability and exact observability of stochastic systems with their applications," *Automatica*, vol. 40, no. 1, pp. 87–94, 2004.
- [21] W. Zhang, Y. Huang, and H. Zhang, "Stochastic H_2/H_∞ control for discrete-time systems with state and disturbance dependent noise," *Automatica*, vol. 43, no. 3, pp. 513–521, 2007.
- [22] Z. Lin, J. Liu, Y. Lin, and W. Zhang, "Nonlinear stochastic passivity, feedback equivalence and global stabilization," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 9, pp. 999–1018, 2012.
- [23] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*, pp. 151–161, Association for Computational Linguistics, July 2011.
- [24] R. Socher, E. Huang, J. Pennington et al., "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Advances in Neural Information Processing Systems*, vol. 24, pp. 801–809, 2011.
- [25] W. Zhang, B.-S. Chen, and C.-S. Tseng, "Robust H_∞ filtering for nonlinear stochastic systems," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, part 1, pp. 589–598, 2005.
- [26] Z. Lin, Y. Lin, and W. Zhang, "A unified design for state and output feedback H_∞ control of nonlinear stochastic Markovian jump systems with state and disturbance-dependent noise," *Automatica*, vol. 45, no. 12, pp. 2955–2962, 2009.
- [27] W. Zhang, X. Lin, and B. Chen, "LaSalle-type theorem and its applications to infinite horizon optimal control of discrete-time nonlinear stochastic systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 250–261, 2017.
- [28] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 683–697, 1992.
- [29] K. A. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02)*, pp. 311–318, July 2002.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

