Zhang JJ, Zhai FF, Zong CQ. A substitution-translation-restoration framework for handling unknown words in statistical machine translation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(5): 907–918 Sept. 2013. DOI 10.1007/s11390-013-1386-5

A Substitution-Translation-Restoration Framework for Handling Unknown Words in Statistical Machine Translation

Jia-Jun Zhang (张家俊), *Member*, *CCF*, Fei-Fei Zhai (翟飞飞)
and Cheng-Qing Zong (宗成庆), *Senior Member*, *CCF*

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

E-mail: {jjzhang, ffzhai, cqzong}@nlpr.ia.ac.cn

Received December 4, 2012; revised May 7, 2013.

Abstract Unknown words are one of the key factors that greatly affect the translation quality. Traditionally, nearly all the related researches focus on obtaining the translation of the unknown words. However, these approaches have two disadvantages. On the one hand, they usually rely on many additional resources such as bilingual web data; on the other hand, they cannot guarantee good reordering and lexical selection of surrounding words. This paper gives a new perspective on handling unknown words in statistical machine translation (SMT). Instead of making great efforts to find the translation of unknown words, we focus on determining the semantic function of the unknown word in the test sentence and keeping the semantic function unchanged in the translation process. In this way, unknown words can help the phrase reordering and lexical selection of their surrounding words even though they still remain untranslated. In order to determine the semantic function of an unknown word, we employ the distributional semantic model and the bidirectional language model. Extensive experiments on both phrase-based and linguistically syntax-based SMT models in Chinese-to-English translation show that our method can substantially improve the translation quality.

Keywords statistical machine translation, distributional semantics, bidirectional language model

1 Introduction

In statistical machine translation (SMT), unknown words are the source language words that are not seen in the training data and thus have no corresponding translations. The current SMT systems either discard the unknown words or copy them literally into the output. It is well known that unknown words are a big hindrance which greatly influences the translation quality. This problem could be especially severe when the available bilingual data is very scarce.

What kinds of negative impacts would the unknown words have? First and at least, we cannot get the meaning of the unknown words in the target language. For instance, using our training data for Chinese-to-English translation on news domain, the Chinese word *诉请* is unknown in the test sentence "... *向(to) 法院(court) 诉请*...", thus we have no idea about the meaning of this word in the English side.

Second, the unknown words can negatively affect the lexical selection and reordering of their surrounding words. Take the same Chinese sentence as an example, if the Chinese verb *诉请* is kept untranslated in the output, we are likely to obtain the wrong phrase reordering *to the court 诉请* while the correct one is *诉请 to the court*.

The conventional solution of the unknown words is to find their translation with additional resources in various ways[1-8]. They use multilingual data, web data or linguistic resources such as WordNet[9] to induce the translation of unknown words. However, most of these studies only address some parts of unknown words, such as named entities[10-11], abbreviations[3-4], compounds[3,12] and morphological variants[13-14]. Many unknown words still remain untouched. Furthermore, for the unknown words handled, their translation may not help the lexical selection and reordering of the surrounding words. The reason is that the translation is obtained from other resources rather than the original bilingual training data from which translation rules and reordering models are learned. For example, even if a translation of the Chinese word *诉请* is obtained (*appeal* for instance) with external resources, the SMT model has no idea about

Regular Paper

Supported by the National High Technology Research and Development 863 Program of China under Grant Nos. 2011AA01A207, 2012AA011101, and 2012AA011102.

the reordering between *to court* and *appeal* because the reordering model is trained without any information about the source word *诉请* and its translation *appeal*.

From the above analysis, we can see that it is very difficult to obtain the correct translation of the unknown words. Furthermore, lexical selection and reordering of surrounding words are not well handled in the previous methods.

In this paper, we take a step back and try to answer the question whether or not we can solve the second problem of the unknown words without translating them. In other words, rather than making efforts to get the translation of the unknown words, we aim to handle the lexical selection and reordering of their surrounding words without any additional resources. Our main idea is based on the following assumption: the lexical selection and reordering of the surrounding words depend on the semantic function of the unknown word. The semantic function of a word is the syntactic and semantic role the word plays in the sentence. Thus the semantic function determines what context the word should take in the source and target languages. In turn, we can say that two words are similar in semantic function if they take the similar context. With the above assumption, to solve the lexical selection and reordering of the surrounding words, we just need to determine the semantic function of the unknown word. Using the context as a bridge, we can denote the semantic function of a word W by another word W' which shares the most similar context to W.

To find an in-vocabulary word having the most similar semantic function to the unknown word, we propose a Substitution-Translation-Restoration (STR) framework which consists of three steps as follows:

Substitution Step[15]. We propose the distributional semantic model and bidirectional language model respectively to find an in-vocabulary word which shares the most similar context to the unknown word. We substitute the found in-vocabulary word for the unknown word.

Translation Step. After substitution, we input the new source language sentence to the SMT system. Then, we obtain the translation output.

Restoration Step. We search the target language word in the output, which is translated by the in-vocabulary word, and replace it back with the unknown word. The unknown words in the final output can still be translated with other approaches.

For example, we have a Chinese sentence "... *为*(*is*) *百分之六 左右*(*about*) ..." in which *百分之六* is an unknown word that means 6%. Using the proposed model, we find that *一半*(*50%*) in the training data takes the most similar semantic function to the unknown word *百分之六*. Then, we replace the unknown word with *一半*(*50%*) and the SMT system yields the translation "... *is about 50%* ..." ("*一半 左右 ||| about 50%*" happens to be a translation rule). At last, we replace *50%* back with the unknown word *百分之六* resulting in "... *is about 百分之六* ...". By doing so, we obtain the correct reordering of the surrounding words and it makes the translation more understandable.

We can see from above that the most important is the substitution step. Thus, it is our focus in this paper. Two approaches are applied in this step: distributional semantic model and bidirectional language model. Experiments on Chinese-to-English translation① show that, with appropriate constraints, these two models can improve the translation quality greatly.

The remainder of this paper is organized as follows. In the next section, we review the related work on the unknown words translation in SMT. In Sections 3 and 4, we present the distributional semantic model and bidirectional language model respectively. The experimental results and detailed analysis are given in Section 5. The last section concludes this paper.

2 Related Work

In SMT community, several approaches have been proposed to deal with the unknown words. Nearly most of the related researches focus on finding the correct translation of the unknown words with external resources. To translate all kinds of unknown words, [2, 16] adopt comparable corpora and web resources to extract translations for each unknown word. Marton *et al.*[5] and Mirkin *et al.*[6] applied paraphrase model and entailment rules to replace unknown words with in-vocabulary synonyms before translation. However, they used either a large set of additional bitexts or manually compiled synonym thesaurus like WordNet. These resources are not available in many languages. Aziz *et al.*[8] applied the active learning to find the replacement of the unknown words, which requires lots of manpower.

More researches address some specific kind of unknown words, such as Named Entities (NEs), compounds and abbreviations. References [10-11, 17] utilize transliteration and web mining techniques with external monolingual and bilingual corpus, comparable data and the Web to find the translation of the NEs. Reference [4] presents an unsupervised approach to finding the full-form representations for the unknown abbreviations. References [12, 14] translate the compound

①Although we conduct experiments on Chinese-to-English translation, our proposed method is independent of specific languages since we do not use any other external resources except the original training data.

unknown words by splitting them into in-vocabulary words or using translation templates. Reference [3] proposes a sublexical translation method to translate Chinese abbreviations and compounds. It first splits the unknown word into sublexical units, such as Chinese characters; then they find the translations of the sublexical units and obtain the translation of the unknown word by combining the translations of the sublexical units. For translating highly inflected languages, German and Turkish for example, several methods[13,18] use morphological analysis and lexical approximation to translate unknown words.

However, almost all of the above studies do not consider the lexical selection and word reordering of the surrounding words when searching the correct translation of the unknown words. Reference [19] addresses the problem of translating numeral and temporal expressions. It uses manually created rules to recognize the numeral/temporal expressions in the training data and replaces them with a special symbol. Consequently, both of the translation rule extraction and reordering model training consider the special symbol. In the decoding time, if numeral or temporal expression is found, it is substituted by the special symbol so that the surrounding words can be handled properly and finally the numeral/temporal expression is translated with the manually written rules. However, it only deals with the numeral/temporal expressions rather than all kinds of the unknown words.

Totally different from all the previous methods, we do not focus on making great efforts to find the translations for the unknown words with huge external resources. Instead, without using any additional resources, we directly address the problem caused by unknown words: poor lexical selection and word reordering of the surrounding words. In our proposed Substitution-Translation-Restoration framework, the translation and restoration steps are easy to implement while the substitution step[15] is the core of this paper. In the next two sections, the distributional semantic model and bidirectional language model are introduced respectively to fulfill this step.

3 Distributional Semantic Model

Distributional semantics[20] approximates semantic meaning of a word with vectors summarizing the contexts where the word occurs. Distributional semantic models (DSM), such as Latent Semantic Analysis (LSA)[21] and Hyperspace Analogue to Language (HAL)[22], have been proven to be successful in tasks that aim at measuring semantic similarity between words, for example, synonym detection and concept clustering[23]. DSM is effective in synonym detection when the corpus is large enough. However, in our task, the training data is limited and the unknown words in the test set are not equipped with rich contexts. Therefore, instead of obtaining the synonym of the unknown words, we take a step back and find the appropriate word which has the most similar semantic function to the unknown word using DSM.

Next, we will elaborate how to construct the DSM for our task and detail how to find the in-vocabulary word which has the most similar semantic function to the unknown word.

3.1 Model Construction

As it is summarized in [20], the construction of the DSM usually includes seven steps: 1) linguistic pre-processing, 2) matrix construction: use term-document or term-term matrix, 3) context calculation: choose structured or unstructured context, 4) interpretation: apply geometric or probabilistic interpretation, 5) feature scaling, 6) normalization, and 7) similarity calculation. In the remainder of this subsection, we will detail how we implement these seven steps.

In the linguistic pre-processing, we first merge the source-side of training data *TD* and evaluation data *ED*[②], resulting in the whole monolingual data *MD*. Then, we segment and POS (part-of-speech) tag the monolingual data *MD*. In this paper, we just use the surface form word as the term and the context unit. The POS will be adopted as a constraint when choosing the most appropriate in-vocabulary word for each unknown word.

For the term-term matrix in our task, each row is a vector denoting the context distribution for a term we concern and each column represents a context term. It is easy to see that both the number of rows and columns are equal to the size of the vocabulary of *MD*. Suppose the size of the vocabulary is N, then the term-term matrix is $N \times N$.

To choose the specific context, a context term is chosen if it occurs within a window of K words around the term we concern. We can distinguish left context from right context so as to make the context in a good structure. Here, we just utilize the unstructured context in order to avoid data sparseness. Different window sizes are tried for finding the best one.

To simplify the similarity calculation, we adopt geometric interpretation and construct a vector $\boldsymbol{V}_{tw}$ for each word tw we concern.

②Combining training data with evaluation data is just for computation efficiency. Evaluation data is not involved when we calculate the context vectors of words in the training data. Unknown words in evaluation data are never seen in the training data and naturally the training data is not involved in the calculation of context vectors of the unknown words.

In $\boldsymbol{V}_{tw}$, the i-th element denotes the distribution probability of the i-th vocabulary word as the context for the word tw. Naturally, we can record the co-occurrence frequency for each context term and use it as the i-th element.

In order to take the frequency of the word we concern and the context word into account, we adopt mutual information to do feature scaling. The pointwise mutual information (PMI) between the word tw and the context word cw is:

$$\begin{aligned} PMI(tw, cw) &= \log \frac{p(tw, cw)}{p(tw)p(cw)} \\ &= \log \frac{f_{tcw}/f_{aw}}{(f_{tw}/f_{aw}) \times (f_{cw}/f_{aw})} \\ &= \log \frac{f_{aw} \times f_{tcw}}{f_{tw} \times f_{cw}}, \end{aligned}$$

where f_{tw} and f_{cw} are the occurrence count of the word tw and the context word cw respectively, f_{tcw} is the co-occurrence count of tw and cw, and f_{aw} is the total occurrence count of all words. Therefore, the distributional context vector $\boldsymbol{V}_{tw} = (PMI(tw, cw_1), \ldots, PMI(tw, cw_N))$.

Finally, we apply the cosine measure to calculate the similarity between two words tw and tw', whose distributional context vectors are $\boldsymbol{V}_{tw}$ and $\boldsymbol{V}_{tw'}$ respectively:

$$\begin{aligned} Sim(tw, tw') &= \cos(tw, tw') \\ &= \frac{\langle \boldsymbol{V}_{tw}, \boldsymbol{V}_{tw'} \rangle}{\|\boldsymbol{V}_{tw}\|_2 \times \|\boldsymbol{V}_{tw'}\|_2} \\ &= \langle \boldsymbol{V}^n_{tw}, \boldsymbol{V}^n_{tw'} \rangle, \end{aligned}$$

in which $\boldsymbol{V}^n_{tw}$ and $\boldsymbol{V}^n_{tw'}$ are L_2-norm of $\boldsymbol{V}_{tw}$ and $\boldsymbol{V}_{tw'}$.

3.2 Search In-Vocabulary Word for Unknown Words

According to the evaluation data and training data, we can easily distinguish unknown words from in-vocabulary words. We denote the unknown words set by UWS and the in-vocabulary words set IWS. For each unknown word UW, our goal is to find the most appropriate word IW^* from IWS so that IW^* has the most similar semantic function to UW. With the similarity function defined above, we can use the following formula to reach our goal:

$$IW^* = \arg\max_{IW} Sim(UW, IW).$$

However, we find that using this formula without any constraint usually cannot obtain good results. Therefore, we require that the resulting in-vocabulary word IW^* should have the consistent part-of-speech with the unknown word UW. Accordingly, the search formula will be:

$$IW^* = \arg\max_{IW \in \{IW' | POS(IW') \cap POS(UW) \neq \varnothing\}} Sim(UW, IW). \quad (1)$$

It should be noted that only some of the found in-vocabulary words using (1) working together with the context of the unknown word can match an entry[③] in the translation phrase table. And if they do so, it will facilitate the lexical selection and word reordering of the surrounding words since the phrase pair entry encodes correct word reordering. For instance, in the example sentence "... 为(*is*) 百分之六 左右(*about*) ...", an in-vocabulary word 一半 is found using (1) for the unknown word 百分之六, and after the replacement the sentence becomes "... 为(*is*) 一半 左右(*about*) ...". The substring 一半 左右 matches an entry *"一半 左右 ||| about 50%"* in the phrase table, and the translation of the substring leads to the correct reordering and word selection of the context. To guarantee good word reordering and lexical selection, we further require that any found in-vocabulary word, when combined with the context of the unknown word, should match an entry in the phrase table.

4 Bidirectional Language Model

Distributional semantic model needs to search through all the in-vocabulary words when calculating context similarity for an unknown word. So, it is not efficient. More importantly, the context modeling does not address the word order of the context and the conditional dependence between them. For example, for the word tw and its context $cw_{l\_4} cw_{l\_3} cw_{l\_2} cw_{l\_1} tw cw_{r\_1} cw_{r\_2} cw_{r\_3} cw_{r\_4}$ with window $K = 4$, all words are treated equally without considering the word position and dependence between each other. It indicates that this model misses a lot of important information.

A question arises that how to use the context more effectively? Considering that the goal of our task is to find the most appropriate in-vocabulary word for the unknown word given the left and right context of the unknown word, the objective function can be formulized as follows:

$$IW^* = \arg\max_{IW} P(IW | cw_{\text{left}}, cw_{\text{right}}).$$

Now, let us focus on $P(IW | cw_{\text{left}}, cw_{\text{right}})$ which models the probability distribution of generating a word

③An entry in the phrase table denotes a phrase pair (a translation rule) which includes two parts: source language phrase and its translation. If the new source language phrase (the substitute combined with the original surrounding words of the unknown word) matches an entry, it means that this new source language phrase is the same as the source-side of the entry.

given the left and right context. However, this probability is difficult to estimate because the condition is too strict. Following the n-gram probability estimation, we have two back-off ways: 1) concerning only the left context $P(IW|cw_{\text{left}})$; 2) concerning only the right context $P(IW|cw_{\text{right}})$. Therefore, we can take a step back and search the in-vocabulary word with the constraint combining these two back-off probabilities:

$$IW^* = \arg\max_{IW} P(IW|cw_{\text{left}})P(IW|cw_{\text{right}}). \quad (2)$$

Obviously, the first back-off probability $P(IW|cw_{\text{left}})$ can be modeled using a forward n-gram probability where $n = K + 1$. Thus, we can just use the conventional n-gram probability estimation method to estimate the back-off probability of generating each in-vocabulary word given the left context. We name this back-off model the *forward language model*.

However, it is not intuitive to see how to estimate the second back-off probability $P(IW|cw_{\text{right}})$. In contrast to the forward language model $P(IW|cw_{\text{left}})$, $P(IW|cw_{\text{right}})$ can be regarded as a *backward language model*. The difficulty lies in how to estimate the backward language model. In practice, the backward language model can be easily estimated through the reversion of the training sentence[24]. Take the following sentence for example:

$$w_1 w_2 \dots w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} \dots w_{m-1} w_m, \quad (3)$$

after reversion, the sentence will be:

$$w_m w_{m-1} \dots w_{i+2} w_{i+1} w_i w_{i-1} w_{i-2} \dots w_2 w_1. \quad (4)$$

If we consider trigram language model, the forward trigram language model $P(w_i|w_{i-1}w_{i-2})$ can be estimated using the original sentence (3) and the backward trigram language model $P(w_i|w_{i+2}w_{i+1})$ can be estimated with the reversed sentence (4). For the backward n-gram language probability calculation, we can use the same strategy to reverse the test string first.

Therefore, we call (2) using the forward and backward n-gram language models the *bidirectional language model*. Like the distributional semantic model, we can impose the same part-of-speech constraints on the objective searching function (2), resulting in:

$$IW^* = \arg\max_{IW \in \{IW' | POS(IW') \cap POS(UW) \neq \varnothing\}} P(IW|cw_{\text{left}}) P(IW|cw_{\text{right}}). \quad (5)$$

Likewise, we can further require the obtained in-vocabulary word from (5), when combined with the context of unknown word, must match a corresponding entry in phrase table. It should be noted that in (2), the forward language model and the backward language model are assigned the same weight. Naturally, we can tune the weights of these two models to achieve the best performance. However, we find that different weights lead to similar results. Thus, we just use the same weights in our experiments.

Compared with distributional semantic model, the bidirectional language model can well model the word order and dependence among context words.

5 Experiments

In this section, we first introduce the experimental setting and some preprocessing. Then, the accuracy of finding the in-vocabulary words sharing the most similar semantic function to the unknown words is evaluated in a manual way. Finally, translation experiments on phrase-based and linguistically syntax-based SMT models are conducted to see whether our proposed methods can improve the translation quality.

5.1 Setup

In this experiment, we use the Chinese-English FBIS④ bilingual corpus to train the translation model. We employe GIZA++⑤ and grow-diag-final-and balance strategy to generate the final symmetric word alignment. We train a 5-gram language model with the target part of the bilingual data and the Xinhua portion of the English Gigaword corpus. NIST MT03 test data is adopted as the development set and NIST MT05 test data is employed as the test set. The statistics of the experimental data are listed in Table 1.

Table 1. Statistics of All Datasets

Dataset	Sentences		Words	
	Chinese	English	Chinese	English
FBIS (bilingual data)	235 489	235 489	7 085 086	9 122 805
Development set	919	919 × 4	24 149	114 056
Test set	1 082	1 082 × 4	29 893	141 695
English Gigaword (monolingual data)	10 912 863		279 096 910	

Note: 919 × 4 means that each source sentence has four reference sentences.

We use the toolkit Urheen⑥ for Chinese word segmentation and POS tagging. Among the 1 082 test sen-

④LDC category is LDC2003E14.

⑤http://code.google.com/p/giza-pp/, Apr. 2012.

⑥http://www.openpr.org.cn/index.php/NLP-Toolkit-for-Natural-Language-Processing/, Apr. 2012. This word segment toolkit[25] combines the merits of the generative model and discriminative model, and performs well when the sentence contains unknown words.

tences, there are totally 796 distinct unknown words. According to the part-of-speech⑦, the count distribution of the unknown words is: (NR, 273), (NN, 272), (CD, 122), (VV, 99), (NT, 14), (AD, 7), (JJ, 5), (OD, 2) and (M, 2).

5.2 Experimental Results on Accuracy of Semantic Functions

The proposed two models aim at finding the most appropriate in-vocabulary word that has the most similar semantic function to the unknown word. However, the models cannot promise correct result for each unknown word. In this subsection, we investigate the accuracy of the two models.

Since there is no automatic method to measure the accuracy of the semantic function between any two words, we ask two native speakers of Chinese to evaluate the accuracy manually and use the average. Moreover, we employ the statistic of Cohen Kappa κ[27-28] to measure the inter-rater agreement for the two annotators. Table 2 and Table 3 give the statistics for the distributional semantic model (DSM) and the bidirectional language model (BLM) respectively, where POS and POS+ Teans denote different constraints. In these two tables, we also show the S-measure[27] which is used to compare the DSM and BLM models against the two raters. Here, we consider the labeling of the two raters as a set of ratings, and the model as a classifier. In this way, S-measure can be applied. The higher the S score becomes, the better the model is. Table 2 shows the results of DSM model with different context window sizes and different constraints. Overall, the accuracy of DSM model is not high. We believe it is because that the DSM is a bag-of-words model and the context of the unknown word in test data is very limited. Specifically, we can see that requiring the found in-vocabulary word should have an entry in the phrase table substantially outperforms the model only with POS constraint. Furthermore, among different context windows, the size of 6 performs best. In a deeper analysis, we have found that the unknown words whose POS are NN and VV are the main reason for the low accuracy.

Table 3 shows the manual results for the BLM model (trigram in both directions) with different constraints. It is easy to see that the accuracy of the BLM model is much better than that of the DSM model. We think this is due to the modeling of context word order and dependence between them in the BLM model. We also notice that the model requiring the found in-vocabulary word should have an entry in the phrase table performs best and achieves the accuracy of 77.6%.

Table 2. Manual Evaluation Results for DSM Model with Different Context Window Sizes and Different Constraints

Window Size	POS			POS+Trans		
	Average Accuracy (%)	κ	S	Average Accuracy (%)	κ	S
4	52.5	0.42	0.51	58.6	0.46	0.53
5	54.4	0.38	0.47	62.8	0.40	0.51
6	62.5	0.34	0.50	69.2	0.48	0.56
7	50.1	0.51	0.53	57.3	0.49	0.57

Table 3. Manual Evaluation Results for BLM Model with Different Constraints

Constraint	Average Accuracy (%)	κ	S
Without POS	68.5	0.52	0.61
With POS	73.9	0.47	0.57
POS+Trans	77.6	0.56	0.63

For the inter-rater agreement, Table 2 and Table 3 show that the two annotators are more likely to agree with each other when we use the BLM model. S-measure also shows that the BLM model is much better than the DSM model.

5.3 Experimental Results on Translation Quality

In this subsection, we evaluate the translation results of the DSM model and the BLM model. To have a better comparison, we apply two translation models: a phrase-based model and a linguistically syntax-based model. We report all the results with case-insensitive BLEU (Bilingual Evaluation Understudy)-4 using shortest length penalty (main metric) and NIST. There are several methods about evaluating learning algorithms[29-30]. Here, we employ pairwise re-sampling approach to perform significance test[29].

5.3.1 Experimental Results on Phrase-Based Model

The translation procedure of the phrase-based SMT model can be viewed as a three-step process: phrase segmentation, translation and reordering.

Fig.1 illustrates a translation example for the phrase-based model. Obviously, if an unknown word exists (Fig.1(b)), the translation result will be much affected. The translation result is "*parts of Europe 遭受 floods hit*" in which the reordering of the surrounding words are wrong. This subsection shows how our proposed models of handing the unknown words improve the phrase-based translation quality.

In phrase-based SMT, a log-linear model is usually utilized to combine multiple sub-model features, such as two phrase translation probabilities, two lexical

⑦The Chinese POS tag set can be referred in [26].

weights, the phrase reordering model, the target language model, the phrase number penalty and the translation length penalty. For the phrase-based translation system, we use the open-source toolkit Moses[31] with its default settings. Minimum-error-rate training[32] is performed on the development set to obtain the weights of the sub-model features.

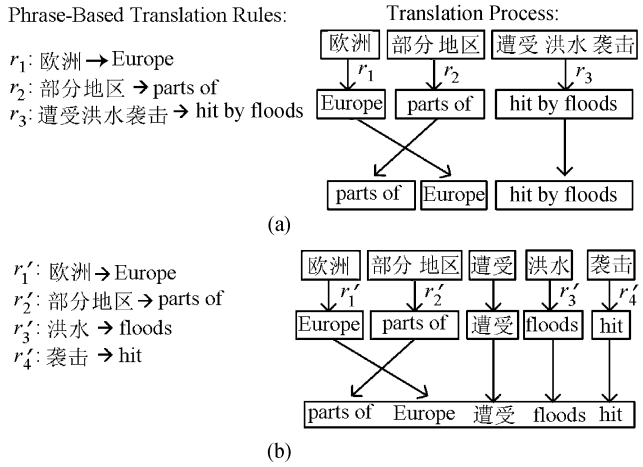

Fig.1. Illustrative example for the phrase-based translation model and the influence of unknown words. (a) Conventional translation process without unknown words. (b) Example containing the unknown word "遭受". The left part presents translation rules and the right part presents the Chinese-to-English translation process.

At first, we conduct a manual analysis to figure out how the unknown word affects the order of its surrounding words. From the sentences containing unknown words, we randomly choose 100 sentences. Among the 100 sentences, the surrounding words of the unknown words in 58 sentences do not need to be reordered and the phrase-based SMT system Moses still gets 23 wrong reorderings. In the remaining 42 sentences which need reordering, Moses gets 34 wrong reorderings. Obviously, the unknown word is an important factor that impacts on the reordering of the surrounding words.

Now, we give the experimental results using our proposed methods. Table 4 gives the translation results using the DSM model with different context window sizes and different constraints. The last line shows the performance of the baseline using default Moses. With only POS constraint, the DSM model with window 4 and 7 even degrades the translation quality. The reason is obvious since Table 2 shows that their accuracy of semantic function is only around 50%. When augmented with translation entry constraint, the model outperforms the baseline in all different window sizes. The model with window 6 performs best and obtains an improvement of 0.42 BLEU score over the baseline.

Table 4. Translation Results for DSM with Different Window Sizes and Constraints

Window Size	BLEU (%)		NIST	
	POS	POS + Trans	POS	POS + Trans
4	29.53	30.02	8.225 4	8.359 2
5	29.86	29.88	8.448 7	8.369 4
6	30.02	30.16	8.429 6	8.391 0
7	29.66	30.01	8.372 4	8.452 8
Baseline	29.74		8.313 9	

Table 5 illustrates the translation results of the BLM model with different constraints. We can see that the bidirectional language model can always obtain better translation quality compared with the baseline. Specifically, the BLM model with the POS constraint significantly outperforms the baseline by 0.54 BLEU score. When enhanced with translation entry constraint, the BLM model achieves the best performance and obtains a statistically significant improvement of 0.64 BLEU score. The results have shown that the BLM model is very effective to handle unknown words in SMT even though the model is relatively simple.

Table 5. Translation Results for BLM with Different Constraints

Constraint	BLEU (%)	NIST
Without POS	29.89	8.388 5
With POS	**30.28**	8.410 8
POS+Trans	**30.38**	8.465 9
Baseline	29.74	8.313 9

Note: Bold figures in the table show that the results are significantly better than the baseline with the level $p < 0.05$. The significance test is done using the approach in [8].

To better show the effectiveness, we further compare our proposed method with another approach which finds a synonym to replace the unknown word before translation. We utilize the Chinese thesaurus TongYiCiCiLin⑧ (extended version) to find the synonyms of the unknown words. The BLEU and NIST scores of this approach are 29.93 and 8.322 7 respectively. It does not perform so well as our method since we find that many unknown words have no synonyms in the thesaurus and some found synonyms lack translation in our bilingual training data.

To have a more comprehensive comparison, we also conduct the experiments with the forward language model and backward language model respectively. Table 6 and Table 7 give the translation results respectively. As demonstrated, both forward language model

⑧http://www.ir-lab.org, July 2011.

and backward language model cannot outperform the bidirectional language model. The results also show that the forward language model performs slightly better than the backward language model. It is consistent with the conclusion drawn by [24] that forward language model is more effective than backward language model for Chinese.

Table 6. Translation Results for Forward Language Model with Different Constraints

Constraint	BLEU (%)	NIST
Without POS	29.65	8.288 2
With POS	29.98	8.390 0
POS+Trans	30.21	8.426 8

Table 7. Translation Results for Backward Language Model with Different Constraints

Constraint	BLEU (%)	NIST
Without POS	29.67	8.318 9
With POS	29.82	8.412 7
POS+Trans	30.15	8.460 2

In order to have a better intuition about the performance improvement, we compare the baseline with our proposed method using bidirectional language model with POS+Trans constraint in three translation examples. Fig.2 illustrates the results.

In the first example, the unknown word in the original Chinese sentence is a number "25%". Without further processing, both of the word selection and reordering in target language are not well performed. All of the source language words are translated one after another and the word "总数" is not translated since the baseline selects the wrong phrase pair (选民 总数, electorate) for translation, making the final English translation hard to understand. Applying the proposed Substitution-Translation-Restoration framework with the bidirectional language model, "一半" is found to replace "25%" before decoding, and we can finally obtain the correct lexical selection and word reordering.

In the second example, the unknown word is a verb "作成". If we copy this unknown word literally into the output, the successive Chinese word "决定" which is a noun is mistakenly converted into an English verb. In contrast, using our proposed method to substitute the in-vocabulary word "作出" for the unknown word "作成", the target language word selection of the successive Chinese noun word "决定" is correct. It makes the translation more understandable.

In the third example, due to the existence of the Chinese unknown word "义演", the translation of the baseline yields the wrong word reordering for the phrase around the Chinese word "的". Fortunately, the proposed Substitution-Translation-Restoration framework leads to the proper word reordering after replacing the unknown word with an appropriate in-vocabulary word. It makes the translation more readable even though the unknown word still keeps untranslated in the final translation.

It should be noted that, if we try other methods to get the target language translation of the unknown word after the restoration step, the translation could be both understandable and readable.

5.3.2 *Unknown Word Translation After Restoration Step*

Sometimes, we can adopt certain methods or external resources to obtain the translation of all the un-

Example 1		
Source Sentence	... 超过 全国 选民 总数 的 *25%* ...	(Unknown Word: *25%*)
Baseline:	... exceeded national electorate *25%* ...	
Substitution:	...超过 全国 选民 总数 的 *一半*...	(Substitution Word: *一半*)
Translation:	... more than half of the total number of voters in the country ...	
Restoration:	... more than *25%* of the total number of voters in the country ...	
Example 2		
Source Sentence	... 内阁 才 *作成* 决定...	(Unknown Word: *作成*)
Baseline:	... the cabinet *作成* decided ...	
Substitution:	... 内阁 才 *作出* 决定 ...	(Substitution Word *作出*)
Translation:	... before the cabinet made the decision ...	
Restoration:	... before the cabinet *作成* the decision ...	
Example 3		
Source Sentence	... *义演* 现场 的 热烈 气氛...	(Unknown Word: *义演*)
Baseline:	... live *义演* and warm atmosphere ...	
Substitution:	... *演习* 现场 的 热烈 气氛...	(Substitution Word *演习*)
Translation:	...the warm atmosphere of the exercise ...	
Restoration:	...the warm atmophere of the *义演* ...	

Fig.2. Comparison examples between the baseline and our proposed method.

known words. In this situation, a question may arise that whether or not our Substitution-Translation-Restoration framework can still outperform the baseline. In this case, the baseline will not encounter any unknown words since each unknown word and its translation becomes a translation rule; while our Substitution-Translation-Restoration framework will become four steps: substitution, translation, restoration and unknown word translation. That is to say, the baseline translates the unknown words in the decoding stage while our method gets the translation of unknown words after the substitution-translation-restoration process.

In this paper, we utilize the phrase-based SMT model to conduct the experiments to verify which method is more effective. Since finding the correct translation of the unknown words is not the concern of this paper, we just resort to Google Translator[⑨] to obtain the English translation of all the 796 distinct Chinese unknown words in the test set. Since the proposed bidirectional language model with POS+Trans constraint performs best in the above experiments, we choose this model in our Substitution-Translation-Restoration framework.

Table 8 shows the experimental results. When comparing the translation quality before and after integrating translation of unknown words, we can obviously see that incorporating unknown word translations in baseline method and our proposed method both significantly outperform the original one. Specifically, for the baseline method, the BLEU score improvement is up to 1.19 points (30.93 vs 29.74). For the Substitution-Translation-Restoration framework with BLM model, the BLEU score gains are 1.14 points (31.52 vs 30.38). If we compare our proposed method integrating unknown word translations with the baseline, we find that it can still obtain a statistical improvement of 0.59 BLEU points (31.52 vs 30.93). The results indicate that our Substitution-Translation-Restoration framework is effective no matter whether or not we integrate the unknown word translations.

Table 8. Translation Performance Comparison Between Baseline and Our Substitution-Translation-Restoration Framework Given the Translations of All the Unknown Words

Method	BLEU (%)	NIST
Baseline (original)	29.74	8.313 9
Baseline (new)	**30.93**	8.821 4
BLM (original)	30.38	8.465 9
BLM (new)	**31.52+**	8.879 3

Note: "original" means that we do not consider the translation of unknown words, while "new" means the translation of unknown words is integrated. Bold figures denote that the new system significantly outperforms the original system with the level $p < 0.01$, and "+" denotes that new BLM system significantly performs better than the new baseline system with the level $p < 0.05$.

5.3.3 *Experimental Results on Linguistically Syntax-Based Model*

For the linguistically syntax-based SMT models, especially for the string-to-tree model, the existence of unknown words greatly degrades the translation performance. The reason is that in the string-to-tree translation model, besides obtaining the translation of a source language sentence, we need to construct a phrase structure tree for the translation synchronously. If an unknown word exists, we fail to decide its translation and more importantly, we have no idea of how to determine the part-of-speech tag of the translation.

Here, we use a string-to-tree example to better illustrate the importance of the unknown words. Informally, the string-to-tree translation model views the translation procedure as a monolingual parsing process with Synchronous Tree Substitution Grammars (STSG). It parses the source sentence with the source-side of the STSG rules and synchronously builds a target phrase structure tree with the target-side of the STSG rules. Fig.3 gives the STSG rules and Fig.4 presents a Chinese-to-English string-to-tree translation example using the rules given in Fig.3. If the Chinese word "警方" is an unknown word, its English translation is unknown accordingly, and furthermore, we cannot figure out which English phrase structure corresponds to the Chinese unknown word. Then, we do not know how to build the high-level tree structure. Should we create a part-of-speech tag or a phrase tag in the target language side for the unknown word? Which tag should we choose? Due to the existence of the unknown word, we have no idea about the answer.

r_1 : NP → (枪手, DT(the) NNS(gunmen)),

r_2 : IN → (被, IN (by)),

r_3 : NP → (警方, DT(the) NN(police)),

r_4 : PP → (x_0x_1, IN : x_0 NP : x_1),

r_5 : VP → (x_0 击毙, VBD(were) VP(VBN(killed) PP:x_0)),

r_6 : S → (x_0x_1, NP : x_0 VP : x_1).

Fig.3. Example string-to-tree STSG translation rules.

Conventionally, we may adopt some heuristic approaches. For example, no matter what the unknown word is, we assume that both part-of-speech tag and phrase tag could be its corresponding target structure.

[⑨] http://translate.google.cn/, October 2012. It translates the unknown words without the context information.

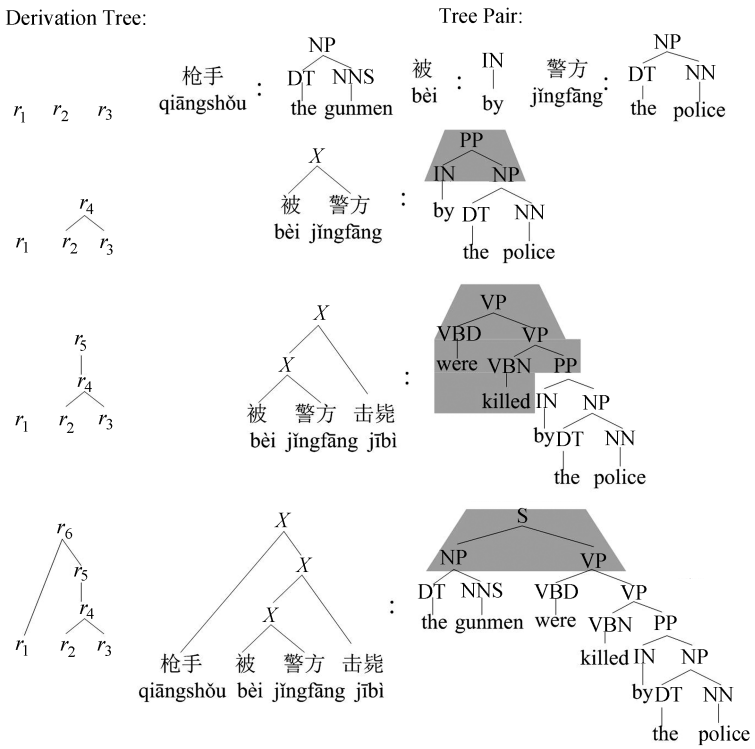

Fig.4. Illustration of string-to-tree translation.

Then, we will choose the tag that can build the best whole tree structure. We name this heuristic *naive heuristic*.

Maybe we can use a smarter heuristic. During STSG rule extraction from the word-aligned and target-side parsed bilingual training data, we first perform part-of-speech tagging for the source sentences, and then we can learn the set of target-side part-of-speech tags and phrase tags which are aligned to each source-side part-of-speech tag. For instance, we may find that the English tag set (NN:0.6, NNS:0.5, NP:0.4, PP:0.2, VP:0.1) is learned for the Chinese part-of-speech tag NN. That is to say, in training data, a Chinese noun is translated into an English noun with a probability of 0.6 and is translated into an English prepositional phrase with a probability of 0.2, and so on. Thus, given a Chinese test sentence with its part-of-speech tags, if an unknown word exists, we can determine the English tag set according to the part-of-speech of the unknown word. Similar to the naïve heuristic, we finally choose a tag from the tag set which can create the best whole tree structure. We name this heuristic *smart heuristic*.

However, the two heuristics are not optimal since they do not consider the context of the unknown words. Applying our proposed Substitution-Translation-Restoration framework, we can successfully bypass choosing the corresponding target tag set as the unknown word does not exists after the substitution step, and moreover we can make the target-side word selection and reordering more reasonable for the context (surrounding words) of the unknown word. Thus, we expect that our method can contribute more to the translation quality. As we have done in the previous subsection, we adopt the bidirectional language model (BLM) with the POS+Trans constraint in the substitution step for the experiment.

We will conduct experiments to compare the two heuristics and our Substitution-Translation-Restoration framework in handling unknown words using string-to-tree SMT model. For the string-to-tree model, we use our in-house implementation according to [33-36]. It is also a log-linear model integrating multiple sub-model features, such as rule translation probability given the root, source-side and target-side, and lexical weights, target language model, and rule number penalty. The feature weights are also tuned with minimum-error-rate training on the development set. The training, development and test data are the same as those in the phrase-based models.

Table 9 shows the results. Thanks to the modeling of the target language syntax, the string-to-tree model can significantly perform better than the phrase-based model Moses (30.33 vs 29.74) with the level $p < 0.05$. If we investigate which method of handling the unknown words leads to better translation performance, we can see from the table that the two heuristics are compara-

ble to each other in translation quality (30.33 vs 30.25). However, when applying our proposed Substitution-Translation-Restoration framework with BLM model, the translation quality can be greatly improved. Our method can obtain an improvement of 0.71 BLEU score points over the naive heuristic and 0.63 BLEU score points over the smart heuristic. This indicates that the proposed Substitution-Translation-Restoration framework is effective for both phrase-based models and linguistically syntax-based models.

Table 9. Translation Results of Different Approaches Dealing with the Unknown Words in String-to-Tree Model

Method	BLEU (%)	NIST
Naive heuristic	30.25	8.472 3
Smart heuristic	30.33	8.449 0
BLM	**30.96**	8.623 5

Note: The bold figure means BLM outperforms the naive heuristic and smart heuristic statistically with the level $p < 0.05$.

6 Conclusions

This paper has presented a Substitution-Translation-Restoration framework to handle the unknown words in statistical machine translation. Instead of trying hard to obtain the translation of the unknown words, this paper has proposed to find the in-vocabulary words that have the most similar semantic function to the unknown words and replace the unknown words with the found in-vocabulary words before translation. By doing this, we can well handle the lexical translation and word reordering for the context of the unknown words during decoding.

Distributional semantic model and bidirectional language model were introduced in the substitution step. Both phrase-based and linguistically syntax-based SMT models were employed to test the effectiveness of our method. In the phrase-based model, we showed that distributional semantic model and the bidirectional language model can both improve the translation quality. Compared with the distributional semantic model, the bidirectional language model performed much better. Moreover, the bidirectional language model also showed its effectiveness in the linguistically syntax-based model.

In the future, we plan to explore new effective models in the substitution step in our Substitution-Translation-Restoration framework.

References

[1] Eck M, Vogel S, Waibel A. Communicating unknown words in machine translation. In *Proc. the 6th LREC*, May 26-June 1, 2008, pp.1542-1547.

[2] Fung P, Cheung P. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *Proc. the 1st EMNLP*, July 2004, pp.57-63.

[3] Huang C, Yen H, Yang P, Huang S, Chang J. Using sublexical translations to handle the OOV problem in machine translation. *ACM Transaction on Asian Language Information Processing*, 2011, 10(3): Article No.16.

[4] Li Z, Yarowsky D. Unsupervised translation induction for Chinese abbreviations using monolingual corpora. In *Proc. the 46th ACL*, June 2008, pp.425-433.

[5] Marton Y, Callison-Burch C, Resnik P. Improved statistical machine translation using monolingually-derived paraphrases. In *Proc. the 6th EMNLP*, August 2009, pp.381-390.

[6] Mirkin S, Specia L, Cancedda N, Dagan I, Dymetman M, Szpektor I. Source-language entailment modeling for translating unknown terms. In *Proc. the 47th ACL*, August 2009, pp.791-799.

[7] Nagata M, Saito T, Suzuki K. Using the web as a bilingual dictionary. In *Proc. the Workshop on Data-Driven Methods in Machine Translation*, July 2001, Vol.14.

[8] Aziz W, Dymetman M, Mirkin S, Specia L, Cancedda N, Dagan I. Learning an expert from human annotations in statistical machine translation: The case of out-of-vocabulary words. *In Proc. the 14th EAMT*, May 2010.

[9] Miller G A. WordNet: A lexical database for English. *Magazine Communications of the ACM*, 1995, 38(11): 39-41.

[10] Knight K, Graehl J. Machine transliteration. In *Proc. the 35th ACL*, July 1997, pp.128-135.

[11] Jiang L, Zhou M, Chien L, Niu C. Named entity translation with web mining and transliteration. In *Proc. the 20th IJCAI*, January 2007, pp.1629-1634.

[12] Tanaka T, Baldwin T. Noun-noun compound machine translation: A feasibility study on shallow processing. In *Proc. the ACL Workshop on Multiword Expressions: Analysis, Acquisition, and Treatment*, July 2003, pp.17-24.

[13] Arora K, Paul M, Sumita E. Translation of unknown words in phrase-based statistical machine translation for languages of rich morphology. In *Proc. the 1st Workshop on Spoken Language Technologies for Under-Resourced Languages*, May 2008, pp.70-75.

[14] Koehn P, Knight K. Empirical methods for compound splitting. In *Proc. the 10th EACL*, April 2003, pp.187-193.

[15] Zhang J, Zhai F, Zong C. Handling unknown words in statistical machine translation from a new perspective. In *Proc. the 1st CCF Conference on NLP&CC*, October 31-November 5, 2012, pp.176-187.

[16] Shao L, Ng H. Mining new word translations from comparable corpora. In *Proc. the 20th COLING*, August 2004, Article No. 618.

[17] Al-Onaizan Y, Knight K. Translating named entities using monolingual and bilingual resources. In *Proc. the 40th ACL*, July 2002, pp.400-408.

[18] Langlais P, Patry A. Translating unknown words by analogical learning. In *Proc. the 4th EMNLP*, June 2007, pp.877-886.

[19] Li H, Duan N, Zhao Y, Liu S, Cui L, Hwang M, Axelrod A, Gao J, Zhang Y, Deng L. The MSRA machine translation system for IWSLT-2010. In *Proc. the 7th IWSLT*, December 2010, pp.135-138.

[20] Evert S. Distributional semantic models. In *Tutorial of NAACL-HLT*, June 2010, http://wordspace.collocations.de/doku.php/course:acl2010:schedule, July 2013.

[21] Landauer T, Dumais S. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 1997, 104(2): 211-240.

[22] Lund K, Burgess C. Producing high-dimensional semantic spaces from lexical cooccurrence. *Behavior Research Methods*, 1996, 28(2): 203-208.
[23] Turney P, Pantel P. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 2010, 37(1): 141-188.
[24] Xiong D, Zhang M, Li H. Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers. In *Proc. the 49th ACL*, June 2011, pp.1288-1297.
[25] Wang K,Zong C, Su K. Integrating generative and discriminative character-based models for Chinese word segmentation. *ACM Transactions on Asian Language Information Processing*, 2012, 11(2): Article No.7.
[26] Xia F. The part-of-speech guidelines for the Penn Chinese treebank (3:0). Technical Report, IRCS Report 00-07, University of Pennsylvania, Oct. 2000.
[27] Shah M. Generalized agreement statistics over fixed group of experts. In *Proc. the ECML PKDD 2011, Part III*, September 2011, pp.191-206.
[28] Cohen J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1): 37-46.
[29] Koehn P. Statistical significance tests for machine translation evaluation. In *Proc. the 1st EMNLP*, July 2004, pp.388-395.
[30] Japkowicz N, Shah M. Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press, March 2011.
[31] Koehn P, Hoang H, Birch A *et al.* Moses: Open source toolkit for statistical machine translation. In *Proc. the 45th ACL*, June 2007, pp.177-180.
[32] Och F J. Minimum error rate training for statistical machine translation. *In Proc. the 41st ACL*, July 2003, pp.160-167.
[33] Galley M, Hopkins M, Knight K, Marcu D. What's in a translation rule? In *Proc. the NAACL*, May 2004, pp.273-280.
[34] Galley M, Graehl J, Knight K, Marcu D, DeNeefe S, Wang W, Thayer I. Scalable inference and training of context-rich syntactic translation models. In *Proc. the 21st COLING and 44th ACL*, July 2006, pp.961-968.
[35] Marcu D, Wang W, Echihabi A, Knight K. SPMT: Statistical machine translation with syntactified target language phrases. In *Proc. the 3rd EMNLP*, July 2006, pp.44-52.
[36] Zhang J, Zhai F, Zong C. Augmenting string-to-tree translation models with fuzzy use of source-side syntax. In *Proc. the 8th EMNLP*, July 2011, pp.204-215.

Jia-Jun Zhang received the B.Sc. degree from Jilin University, Changchun, in 2006 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2011, both in computer science. Since 2011, he has been with the National Laboratory of Pattern Recognition, which is a part of the Institute of Automation, Chinese Academy of Sciences, Beijing, as an assistant professor. His research interests include statistical machine translation, natural language processing.

Fei-Fei Zhai received the B.Sc. degree in 2009 from Beijing Jiaotong University. He is a Ph.D. candidate of the Institute of Automation, Chinese Academy of Science. His current research direction is machine translation.

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