Handling Unknown Words in Statistical Machine Translation from a New Perspective

Jiajun Zhang, Feifei Zhai, and Chengqing Zong

NLPR, Institute of Automation Chinese Academy of Sciences, Beijing, China
{jjzhang,ffzhai,cqzong}@nlpr.ia.ac.cn

Abstract. Unknown words are one of the key factors which drastically impact the translation quality. Traditionally, nearly all the related research work focus on obtaining the translation of the unknown words in different ways. In this paper, we propose a new perspective to handle unknown words in statistical machine translation. Instead of trying great effort to find the translation of unknown words, this paper focuses on determining the semantic function the unknown words serve as in the test sentence and keeping the semantic function unchanged in the translation process. In this way, unknown words will 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 each unknown word, this paper employs the distributional semantic model and the bidirectional language model. Extensive experiments on Chinese-to-English translation show that our methods can substantially improve the translation quality.

Keywords: statistical machine translation, unknown words, 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 target 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 becomes very scarce.

A question arises that what kinds of negative impacts would the unknown words cause? First and at least, we cannot get the meaning of the unknown words in the target language. For instance, using our small-scale training data for Chinese-to-English translation, 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

M. Zhou et al. (Eds.): NLPCC 2012, CCIS 333, pp. 176–187, 2012.

obtain the wrong phrase reordering *to the court诉请* while the correct one is *诉请 to the court*.

The conventional solution of unknown words is to find their target correspondence with additional resources in different ways[3,5,6,13,16,18,19]. They use multilingual data, web data or linguistic resources such as WordNet[17] to induce the translation of unknown words. By doing this, they hope to solve the above two problems perfectly. However, most of these work only address some part of unknown words, such as Named Entities[7,15], abbreviations[6,13], compounds[6] and morphological variants[2,9]. Therefore, 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 because the translation is obtained from other resources rather than the original bilingual training data from which translation rules and reordering models are acquired. For example, even if a translation of the Chinese word *诉请* is found (*appeal* for instance) with additional resources, SMT systems have no idea about the reordering between *to court* and *appeal* because reordering model is trained without any information about the source word *诉请* and its translation *appeal*.

From the above analysis, we know that finding translation of unknown words needs additional resources and only some part of them can be well handled. Moreover, the translations obtained from additional resources are not consistent with the original training data and fail to guide the lexical selection and reordering of surrounding words. As we can see that, it is very difficult to obtain the correct translation of unknown words and meantime well guide lexical selection and reordering of surrounding words.

In this paper, we take a step back and try to answer the question whether we can solve the second problem of the unknown words without translating them. In other words, rather than trying hard to get the target translation of unknown words, we aim to handle the lexical selection and reordering of their surrounding words quite well 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 the unknown word servers as. The **semantic function** of a word means the syntactic and semantic role the word plays in the sentence, and thus the semantic function determines what context the word should take in source and target language. In turn, we can say that two words are similar in semantic function if they often take the similar context. With 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 with *W*. Therefore, the central idea of this paper is to find a known word having the most similar semantic function with the unknown word. Our method consists of three steps as follows:

First, we use the distributional semantic model and bidirectional language model respectively to find an in-vocabulary word in original training data, which shares the most similar context with the unknown word.

Second, we replace the unknown word with the found in-vocabulary word and input the test sentence to the SMT system. Then, we obtain the translation output.

Third, we find 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 handled with other approaches which aim to get their translations.

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 in fact means 6%. Using the proposed model, we find that *一半(50%)* in the training data takes the most similar context with the unknown word *百分之六*. Then, we replace the unknown word with *一半(50%)* and the example sentence yields the translation *"… is about 50% …"* since there is an entry *"一半 左右 ||| about 50%"* in the translation phrase table. At last, we replace *50%* back with the unknown word *百分之六* resulting *"… is about 百分之六 …."* It is easy to see that we obtain the correct reordering of the surrounding words and it makes the translation more understandable.

We can see from the method that the first step is the most important. Thus, it is our focus in this paper. We propose two approaches in the framework: distributional semantic model and bidirectional language model. Experiments show that, with appropriate constraints, the two models can find the in-vocabulary words sharing similar semantic function with the unknown words and can much improve translation quality as well.

2 Related Work

In SMT community, several approaches have been proposed to deal with the unknown words. Nearly most of the related research work focus on finding the correct translation of the unknown words with external resources. To translate all kinds of unknown words, [5,20] adopted comparable corpora and web resources to extract translations for each unknown word; [16,18] applied paraphrase model and entailment rules to replace unknown words with in-vocabulary words using large set of additional bitexts or manually compiled synonym thesaurus WordNet. More research works address some specific kind of unknown words, such as Named Entities (NEs), compounds and abbreviations. [1,7,15] utilized transliteration and web mining techniques with external monolingual and bilingual corpus, comparable data and the web to find the translation of the NEs. [13] presented an unsupervised approach to find the full-form representations for the unknown abbreviations. [9] translated the compound unknown words by splitting them into in-vocabulary words or using translation templates. [6] proposed a sublexical translation method to translate Chinese abbreviations and compounds. For translating highly inflected languages, several methods[2,11] used morphological analysis and lexical approximation to translate unknown words. However, almost all of the above works did not consider the lexical selection and word reordering of the surrounding words when searching the correct translation of the unknown words.

[12] addressed the problem of translating numeral and temporal expressions. They used manually created rules to recognize the numeral/temporal expressions in the training data and replaced 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, they only deal with numeral/temporal expressions rather than any kind of unknown words.

Totally different from all the previous methods, we do not focus on trying great effort to find translations for the unknown words with huge external resources. Instead, we address the problem of the lexical selection and word reordering of the surrounding words caused by unknown words. In this paper, we consider all kinds of the unknown words and apply the distributional semantic model and the bidirectional language model to fulfill this task without any additional resources.

3 Distributional Semantic Model

Distributional semantics[4] approximates semantic meaning of a word with vectors summarizing the contexts where the word occurs. Distributional semantic models (DSM), such as LSA[10] and HAL[14], have proven to be successful in tasks that aim at measuring semantic similarity between words, for example, synonym detection and concept clustering[21]. DSM is effective to 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 to find the most appropriate word which has the similar semantic function with the unknown word with DSM.

Next, we 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 with the unknown word.

3.1 Model Construction

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

In the linguistic pre-processing, we first merge the source-side of training data ***TD*** and evaluation data ***ED***, resulting the whole monolingual data ***MD***. Then, we segment and POS (part-of-speech) tagging the monolingual data ***MD***. In this paper, we just use the surface form ***word*** as the target 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.

After pre-processing, we construct a term-term matrix for our task. In the matrix, each row is a vector denoting the context distribution for a target term and each column represents a context term. It is easy to see that both the number of rows and columns equals to the size of the vocabulary of ***MD***. Suppose the size of vocabulary is N, then the term-term matrix is $N \times N$.

Then, we need to choose the specific context. In our work, each context term is chosen if it occurs within a window of *K* words around the target term. We can distinguish left context from right context so as to make the context structured. Here, we just utilize the unstructured context in order to avoid data sparseness. We will try different window *K*s in the experiments to find the best one.

To apply the simple similarity calculation, we adopt geometric interpretation and consider the vector for a target word as a point in the vector space. In the following two steps, we detail how to construct the vector V_{tw} for each target word *tw*.

For a vector V_{tw}, the *ith* element denotes the distribution probability of the *ith* vocabulary word as the context for the target word *tw*. Naturally, we can record the co-occurrence frequency for each context term and use it as the *ith* element. In order to take the frequency of target word and context word into account, we adopt the association measures mutual information to do feature scaling. Suppose the occurrence count of target word *tw* and context word *cw* is f_{tw} and f_{cw}, the co-occurrence count of *tw* and *cw* is f_{fcw}, and the total occurrence count of all words is f_{aw}. Thus, the Pointwise Mutual Information (PMI) between *tw* and *cw* is:

$$\begin{aligned} PMI(tw, cw) &= \log \frac{p(tw, cw)}{p(tw)\, p(cw)} \\ &= \log \frac{f_{aw} \times f_{tcw}}{f_{tw} \times f_{cw}} \end{aligned} \tag{1}$$

Therefore, the distributional context vector V_{tw} be $V_{tw} = \left(PMI(tw, cw_1), \cdots, PMI(tw, cw_N) \right)$.

Then, we normalize each vector V_{tw} by its L_2-norm, yielding the normalized vector V_{tw}^n.

Finally, we apply the cosine measure to calculate the similarity between two target words *tw* and *tw'*, whose distributional context vectors are V_{tw}^n and $V_{tw'}^n$ respectively:

$$\begin{aligned} Sim(tw, tw') &= \cos(tw, tw') \\ &= \frac{\left\langle V_{tw}^n, V_{tw'}^n \right\rangle}{\left\| V_{tw}^n \right\|_2 \times \left\| V_{tw'}^n \right\|_2} \\ &= \left\langle V_{tw}^n, V_{tw'}^n \right\rangle \end{aligned} \tag{2}$$

3.2 In-vocabulary Word Search for Unknown Words

According to the evaluation data and training data, we can easily distinguish unknown words from in-vocabulary words. Assume that the unknown words set is *UWS* and the in-vocabulary words set is *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 with *UW*. With the similarity function defined above, we can use the following formula to meet our goal:

$$IW^* = \arg\max_{IW} Sim(UW, IW) \tag{3}$$

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:

$$IW^* = \arg\max_{IW \in \{IW' | POS(IW') \cap POS(UW) \neq \Phi\}} Sim(UW, IW) \tag{4}$$

It should be noted that our final purpose is to improve the translation quality, but not all of the found in-vocabulary words using formula (4) can guarantee good translation with the context of the unknown word since they usually lack the corresponding phrase entry in the translation phrase table. Thus, if the found in-vocabulary word combining the context of the unknown word matches an entry in phrase table, it will facilitate the lexical selection and word reordering of the surrounding words. For instance, in the example sentence *"…* 为(is) 百分之六 左右*(about) …"*, an in-vocabulary word 一半 is found using (4) for the unknown word百分之六, and after replacement the sentence becomes *"…* 为(is) 一半 左右*(about) …"*. If there exists a substring一半 左右 has an entry "一半 左右 ||| *about 50%"* in the phrase table, it leads to the correct reordering and word selection of the context. Therefore, in order to guarantee good word reordering and lexical selection, we can further require that any found in-vocabulary word, which combines the context of the unknown word, should have an entry in phrase table.

4 Bidirectional Language Model

In distributional semantic models, the context modeling does not address the word order of the context and the conditional dependence between them. For example, if the context and target word is $cw_{l\_4}\ cw_{l\_3}\ cw_{l\_2}\ cw_{l\_1}\ tw\ cw_{r\_1}\ cw_{r\_2}\ cw_{r\_3}\ cw_{r\_4}$ with window *K*=4, any word appearing in the window of the target word will be treated equally without considering the word position. As a result, this model misses a lot of important information.

A question arises that how to use the context more effectively? In theory, 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. It can be formulized as follows:

$$IW^* = \arg\max_{IW} P\left(IW \mid cw_{left}, cw_{right}\right) \tag{5}$$

Now, let us focus on $P\left(IW \mid cw_{left}, cw_{right}\right)$ which models the probability distribution of generating a word 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 backoff ways for probability estimation: 1) back off to concerning only the left context $P\left(IW \mid cw_{left}\right)$; 2) back off to concerning only the right context $P\left(IW \mid cw_{right}\right)$. Then, we can take a step back and search the in-vocabulary word with the constraint combining the two backoff probabilities:

$$IW^* = \arg\max_{IW} P\left(IW \mid cw_{left}\right) P\left(IW \mid cw_{right}\right) \tag{6}$$

It is easy to see that the first backoff probability $P\left(IW \mid cw_{left}\right)$ can be modeled using a forward *n*-gram probability where *n* equals to the context window *K* plus one. Thus, we can just use the conventional *n*-gram probability estimation method to estimate the backoff probability of generating each in-vocabulary word given the left context. We name this backoff model the *forward language model*.

However, it is not intuitive to see how to estimate the second backoff probability $P\left(IW \mid cw_{right}\right)$. In contrast to the forward language model $P\left(IW \mid cw_{left}\right)$, $P\left(IW \mid cw_{right}\right)$ 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[22]. Take the following sentence for example:

$$w_1\ w_2 \cdots w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} \cdots w_{m-1}\ w_m \tag{7}$$

After reversion, the sentence will be:

$$w_m\ w_{m-1} \cdots w_{i+2} w_{i+1} w_i w_{i-1} w_{i-2} \cdots w_2\ w_1 \tag{8}$$

If we consider trigram language model, the forward trigram language model $p\left(w_i \mid w_{i-1} w_{i-2}\right)$ can be estimated using the original sentence (7) and the backward trigram language model $p\left(w_i \mid w_{i+2} w_{i+1}\right)$ can be estimated with the reversed sentence (8). During the backward *n*-gram language probability calculation, we can apply the same way to first reverse the test string.

Therefore, we call the formula (6) using the forward and backward *n*-gram language model the *bidirectional language model*. Like the distributional semantic model, we can impose the same part-of-speech constraints on the objective searching function (6) resulting the following formula:

$$IW^* = \underset{IW \in \{IW' \mid POS(IW') \cap POS(UW) \neq \Phi\}}{\arg\max} P\left(IW \mid cw_{left}\right) P\left(IW \mid cw_{right}\right) \tag{9}$$

Likewise, we can further require the obtained in-vocabulary word from (9) combing the context of unknown word must have a corresponding entry in phrase table.

Compared with distributional semantic model, the bidirectional language model can well model the word order and dependence among context words. In the following section, we first analyze how often we can find the correct in-vocabulary word sharing the same semantic function with the unknown word. Then, we substitute the found in-vocabulary words for the unknown words in test set and evaluate the effectiveness of these two models in SMT.

5 Experiments

5.1 Set-Up

Since the application environment of our model supposes the training data is relatively scarce[1], we use the relatively small data set to test our proposed models. In this experiment, we used the Chinese-English FBIS bilingual corpus consisting of 236K sentence pairs with 7.1M Chinese words and 9.1M English words. We employed GIZA++ and grow-diag-final-and balance strategy to generate the final symmetric word alignment. We trained 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. We used the open-source toolkit Urheen[2] for Chinese word segmentation and POS tagging. POS of unknown words are predicted by the MaxEnt model. The test set consists of 1082 sentences, and there are totally 796 distinct unknown words. According to the part-of-speech, 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

5.2.1 Accuracy of Semantic Functions

The proposed two models aim at finding the most appropriate in-vocabulary word that has the most similar semantic function with the unknown word. However, the models

[1] Our method can be applied in all the cases where there are unknown words in the test sentences. However, if the training data is relatively scarce and more unknown words exists, the proposed approach can achieve bigger improvements. Note that the training data cannot be too small so that most unknown words cannot find semantic similar in-vocabulary words.

[2] `http://www.openpr.org.cn/index.php/NLP-Toolkit-for-Natural-Language-Processing/`

cannot promise correct result for each unknown word. In this section, we investigate the accuracy of the two models respectively.

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. Table 1 and 2 give the statistics for the distributional semantic model (DSM) and the bidirectional language model (BLM). Table 1 shows the results of DSM model with different context window size and different constraints. Overall, the accuracy of DSM model is not high. We believe that it is because the context of the unknown word in test data is limited and the training data is not large enough. Specifically, we can see that requiring the found in-vocabulary word should have an entry in phrase table substantially outperforms the model only with POS constraint. Furthermore, among different context windows, the size of 6 performs best. In a deep analysis, we have found that the unknown words whose POS are NN and VV are the main reason for the low accuracy.

Table 2 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 phrase table performs best and achieves the accuracy 77.6%.

Table 1. Manual evaluation results for DSM model with different context window size and different constraints (only POS constrained and POS plus translation entry constrained)

window size	POS (accuracy %)	POS+Trans (accuracy%)
4	52.5	58.6
5	54.4	62.8
6	62.5	69.2
7	50.1	57.3

Table 2. Manual evaluation results for BLM model with different constraints

constraint	Accuracy (%)
without POS	68.5
with POS	73.9
POS+Trans	77.6

5.2.2 Translation Results

In this section, we evaluate the translation results of the DSM model and the BLM model. We use the open-source toolkit Moses to conduct all the experiments. We report all the results with case-insensitive BLEU-4 using shortest length penalty (main metric) and NIST. We employ re-sampling approach to perform statistical significance test [8].

Table 3 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 1 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 0.42 BLEU score over the baseline.

Table 4 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. And when enhanced with translation entry constraint, the BLM model achieves the best performance and obtains an 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 3. Translation results for DSM with different window sizes and constraints

window size	BLEU (%) POS	BLEU(%) POS+Trans	NIST POS	NIST POS+Trans
4	29.53	30.02	8.2254	8.3592
5	29.86	29.88	8.4487	8.3694
6	30.02	30.16	8.4296	8.3910
7	29.66	30.01	8.3724	8.4528
baseline	29.74		8.3139	

Table 4. Translation results for BLM with different constraints

constraint	BLEU (%)	NIST
without pos	29.89	8.3885
with pos	**30.28**	8.4108
pos+trans	**30.38**	8.4659
baseline	29.74	8.3139

To have a comprehensive comparison, we have also conducted the experiments with the forward language model and backward language model respectively. Table 5 and 6 give the translation results. The results in tables show that both forward language model and backward language model cannot outperform the bidirectional language model. The results also show that the forward language model performs better

Table 5. Translation results for forward language model with different constraints

constraint	BLEU (%)	NIST
without pos	29.65	8.2882
with pos	29.98	8.3900
pos+trans	30.21	8.4268

Table 6. Translation results for backward language model with different constraints

constraint	BLEU (%)	NIST
without pos	29.67	8.3189
with pos	29.82	8.4127
pos+trans	30.15	8.4602

than the backward language model. It is consistent with the conclusion drawn by [22] that forward language model is more effective than backward language model for Chinese.

6 Conclusion and Future Work

This paper presents a new idea to handle the unknown words in statistical machine translation from a new perspective. Instead of trying hard to obtain the translation of the unknown words, this paper have proposed two new models to find the in-vocabulary words that have the most similar semantic function with the unknown words and replace the unknown words with the found in-vocabulary words before translation. Thus, by doing this, we can well handle the lexical translation and word reordering for the context of the unknown words during decoding. The experimental results show that the proposed distributional semantic model and the bidirectional language model can both improve the translation quality. Compared with the distributional semantic model, the bidirectional language model performs much better. In the future, we plan to explore more features to handle unknown words in SMT even better. Furthermore, we are going to figure out what is the case if a source-side monolingual large corpus is used for the distributional semantic model.

Acknowledgement. We would like to thank the anonymous reviewers for their valuable comments. The research work has been partially funded by the Natural Science Foundation of China under Grant No. 60975053 and 61003160 and also supported by Hi-Tech Research and Development Program ("863" Program) of China under Grant No. 2011AA01A207.

References

1. Al-Onaizan, Y., Knight, K.: Translating named entities using monolingual and bilingual resources. In: Proc. of ACL 2002 (2002)
2. Arora, K., Paul, M., Sumita, E.: Translation of unknown words in phrase-based statistical machine translation for languages of rich morphology. In: Proceedings of the Workshop on Spoken Language Technologies for Under-Resourced Languages, SLTU 2008 (2008)
3. Eck, M., Vogel, S., Waibel, A.: Communicating unknown words in machine translation. In: Proc. of LREC 2008 (2008)
4. Evert. Distributional Semantic Models. In: Tutorial at NAACL-HLT 2010 (2010)

5. Fung, P.N., Cheung, P.: Mining very-non-parallel corpora: parallel sentence and lexicon extraction via bootstrapping and EM. In: Proc. of EMNLP 2004 (2004)
6. 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)
7. Knight, K., Graehl, J.: Machine transliteration. In: Proc. of EACL 1997 (1997)
8. Koehn, P.: Statistical significance tests for machine translation evaluation. In: Proc. of EMNLP 2004 (2004)
9. Koehn, P., Knight, K.: Empirical methods for compound splitting. In: Proc. of EACL 2003 (2003)
10. Landauer, T., Dumais, S.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review 104(2), 211–240 (1997)
11. Langlais, P., Patry, A.: Translating unknown words by analogical learning. In: Proc. of EMNLP 2007 (2007)
12. 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. of IWSLT 2010 (2010)
13. Li, Z., Yarowsky, D.: Unsupervised translation induction for Chinese abbreviations using monolingual corpora. In: Proc. of ACL 2008 (2008)
14. Lund, K., Burgess, C.: Producing high-dimensional semantic spaces from lexical cooccurrence. Behavior Research Methods 28, 203–208 (1996)
15. Jiang, L., Zhou, M., Chien, L., Niu, C.: Named entity translation with web mining and transliteration. In: Proc. of IJCAI 2007 (2007)
16. Marton, Y., Callison-Burch, C., Resnik, P.: Improved statistical machine translation using monolingually-derived paraphrases. In: Proc. of EMNLP 2009 (2009)
17. Miller, G.A.: WordNet: A lexical database for English. Comm. ACM 38, 11 (1995)
18. Mirkin, S., Specia, L., Cancedda, N., Dagan, I., Dymetman, M., Szpektor, I.: Source language entailment modeling for translating unknown terms. In: Proc. of ACL-IJCNLP 2009 (2009)
19. Nagata, M., Saito, T., Suzuki, K.: Using the Web as a bilingual dictionary. In: Proceedings of the ACL Workshop on Data-Driven Methods in Machine Translation (2001)
20. Shao, L., Ng, H.: Mining new word translations from comparable corpora. In: Proc. of COLING 2004 (2004)
21. Turney, P., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research 37, 141–188 (2010)
22. Xiong, D., Zhang, M., Li, H.: Enhancing Language Models in Statistical Machine Translation with Backward N-grams and Mutual Information Triggers. In: Proc. of ACL 2011 (2011)