

# Terminology Translation Error Identification and Correction

Mengyi Liu, Jian Tang, Yu Hong<sup>(✉)</sup>, and Jianmin Yao

School of Computer Science and Technology, Soochow University,  
Suzhou 215006, Jiangsu, China  
mengyiliu22@gmail.com, JohnnyTang1120@gmail.com,  
tianxianer@gmail.com, jyao@suda.edu.cn

**Abstract.** Statistical machine translation (SMT) system requires homogeneous training data in order to get domain-sensitive terminology translations. If the data is multi-domain mixed, it is difficult for SMT system to learn translation probability of context-sensitive terminology. However, terminology translation is important for SMT. The previous work mainly focuses on integrating terminology into machine translation systems and heavily relies on domain terminology resources. In this paper, we propose a back translation based method to identify terminology translation errors from SMT outputs and automatically suggest a better translation. Our approach is simple with no external resources and can be applied to any type of SMT system. We use three metrics: tree-edit distance, sentence semantic similarity and language model perplexity to measure the quality of back translation. Experimental results illustrate that our method improves performance on both weak and strong SMT systems, yielding a precision of 0.48% and 1.51% respectively.

**Keywords:** Statistical machine translation · Domain terminology · Post-processing · Back translation

## 1 Introduction

In general, the performance of the SMT heavily relies on the scale and quality of the training corpora [1]. High-quality and large-scale corpora tends to include richer linguistic phenomena. As a result, the training effect of the statistical model (translation model, language model, and reordering model) in translation system will be improved.

However, applying a generic SMT system to technical documents often leads to wrong results, especially in the translation of domain-specific terminology. This is mostly due to the lack of domain-specific parallel data from which the SMT system can learn translation knowledge. The importance of domain-specific terminology for SMT has been mentioned in several previous work [2, 3]. Most of the work handles the case how to integrate the terminology tightly into the translation system. This requires not only a large amount of in-domain parallel corpora which is often difficult to obtain, especially for low-resourced domains or languages, but also a good expertise in SMT. We look upon the problem from a different perspective where we post-process the

terminology translation instead of modifying the model. We propose a back translation based method to identify the terminology translation errors and suggest a better translation.

Given a sentence, machine translation system will not output an appropriate translation unless the sentence is logical, according with common sense and contextual semantic consistent. In order to facilitate the understanding of the above linguistic phenomena, two pairs of translation examples are given below (Table 1).

**Table 1.** Two pairs of translation examples

<p><b>Source:</b> 智能工厂用传感器和<u>执行器</u>赢得管理操作和知识驱动优化的实时信息。</p> <p><b>Target:</b> <i>Intelligent plants use sensors and <u>actuators</u> to gain real-time information on management operations and knowledge-driven optimization.</i></p> <p>[Sample1]</p>
<p><b>Source:</b> 智能工厂用传感器和<u>演员</u>赢得管理操作和知识驱动优化的实时信息。</p> <p><b>Target:</b> <i>Intelligent factory with sensors and <u>actors</u> to win management operations and knowledge-driven optimization of real-time information.</i></p> <p>[Sample2]</p>

The source sentence in sample1 is normal statements, smooth and fluent on the whole; but in sample2 the source sentence is abnormal statements, phrase “*actor*” is contextual semantic inconsistent obviously. We use Google Translator<sup>1</sup> to translate two source language sentences, and two translation results show difference in syntactic structure and semantic. In the two source sentences, phrases “管理操作” and “知识驱动优化” are used to modify the phrase “实时信息”. From the target sentence in sample1, we can see that phrases “*management operations*” and “*knowledge-driven optimization*” are used to modify the phrase “*real-time information*”, the same as source sentence. But in sample2’s target sentence, phrase “*real-time information*” is used to modify “*knowledge-driven optimization*”, which is deviated from the meaning expressed by the source sentence. We further analyze this linguistic phenomenon and consider this is resulted from the translation mechanism. The system has translated “演员” as “*actors*”, then it prefers “*win management operations*” as next translation rather than “*gain real-time information*” according with comprehensive score (language model et al.).

As can be seen from the above analysis, the irrationality of individual phrase in a sentence can affect the translation of the whole sentence. If the irrational element in the sentence is a term, this phenomenon will become more obvious. The reason for this is

<sup>1</sup> <http://translate.google.cn>.

that term conveys concepts of a text, term translation becomes crucial when the text is translated from its original language to another language [4].

In this paper, we aim to propose a method to identify terminology translation errors of the SMT outputs and suggest a better translation. Compared with integrating terminology into SMT models and building a sophisticated system, our method is simple and do not rely on domain resources. Our method is based on back translation, and we propose three metrics to measure the quality of back translation: (1) tree-edit distance; (2) sentence semantic similarity; (3) language model perplexity. Experimental results illustrate that they are all able to achieve improvements of precision on both weak and strong translation systems.

The remainder of the paper is organized as follows. Section 2 overviews the related work. We present the methodology and detail the metrics in Sect. 3. Section 4 shows the experimental settings and results. Section 5 draws conclusions and describes the future work.

## 2 Related Work

In this section, we briefly introduce related work and highlight the differences between our work and previous studies.

There has been a growing interest for terminology integration into SMT models recently. [5] investigate that bilingual terms are important for domain adaptation of machine translation. Direct integration of terminology into the SMT model has been considered, either by extending SMT training data [2], or via adding an additional term indicator feature into the translation model [3, 5]. [6] propose a binary feature to indicate whether a bilingual phrase contains a term pair. [4] investigate three issues of term translation in the context of document-informed SMT and integrate the three models into hierarchical phrase-based SMT. However, none of the above is possible when we deal with an external black-box SMT system.

[7] employ bilingual term bank as a dictionary and propose a post-processing step for a SMT system, where a wrongly translated term is replaced with a user-provided term translation. [8] propose a demonstration of a multilingual terminology verification/correction service, which detects the wrongly translated terms and suggest a better translation of the terms.

Our work is also related to machine translation error identification. [9] combine syntax feature, vocabulary feature and word posterior probability feature, which are extracted based on LG parsing, and use the binary classifier based on Maximum Entropy Model to predict the label of each word in machine translation. [10] rely on a random forest classifier and 16 features to predict the label of a word. [11] train two classifier models by using bidirectional long short-term memory recurrent neural networks and CRF to complete word level QE Task.

Our work departs from the previous work in two major respects.

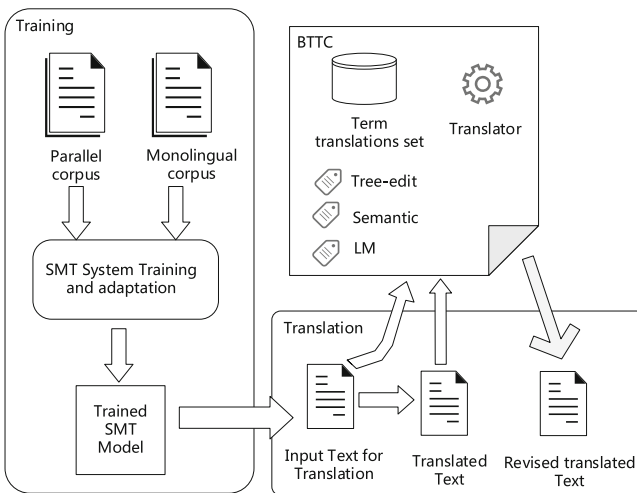
- We focus on the terminology translation error identification and correction, and our method do not rely on external resources such as bilingual domain-specific terminology. This can be seen as post-editing focused on domain terminology.
- Our method is based on back translation, so we just need to compare the same language. This can avoid crossing-language comparison which is complicated.

### 3 Methodology

We propose a method to identify terminology translation errors and automatically suggest better translations. First of all, we present the methodological framework. Then we introduce the crucial part of comparing back translation and original sentence. Finally, we list preprocessing methods for collecting and processing raw data.

#### 3.1 Back Translation Based Terminology-Checking Method

The method proposed in this paper does not modify the model of the translation system, but is used as the post processing of the existing translation system. Figure 1 shows the framework of back translation based terminology-checking method (BTTC).



**Fig. 1.** Framework of BTTC

The left of the framework is the initial SMT system. Model training phase includes phrase table generation, translation model training, reordering model training, and language model training, et al. When these models have been trained, they are combined in a log-linear model. To obtain the best translation  $\hat{e}$  of the source sentence  $f$ ,

log-linear model uses the following equation, in which  $h_m$  and  $\lambda_m$  denote the  $m$ th feature and weight.

$$\begin{aligned}\hat{e} &= \arg \max_e p(e|f) \\ &= \arg \max_e \sum_{m=1}^M \lambda_m h_m(e, f)\end{aligned}\quad (1)$$

Once we obtain a trained SMT system, given a sentence containing terminology, we can translate it into target language. The terminology translation may be correct or wrong and we don't know. To solve this problem, we propose a post-edit processing which contains several steps as follows:

- Locating the terminology translation. To identify the terminology translation errors, the first step is locating its position in the target sentence. Fortunately, we have access to the internal sub-phrase alignment provided by Moses<sup>2</sup>, thus we know the exact location of the terminology translation. We just need to add parameters “-print-alignment-info” when decoding. Specific examples are shown below (Table 2):

**Table 2.** An example of internal sub-phrase alignments

**Source:** *It has great influence on the time complexity of the join algorithms to locate positions on tertiary storage [16 17].*

**Target:** 它可以对时间复杂性影响连接定位算法职位高等教育仓库. ||| 0-0 0-1 6-3 7-4 3-5 10-6 11-8 13-7 14-9 16-10 17-11 18-12

The position of phrase “*tertiary storage*” in the source sentence is 16 and 17, and we can know the position of its translation in target sentence is 10 and 11 according to the alignment information, exactly the phrase “*高等教育仓库*”.

- Replacing terminology translation with other translations. The terminology we marked in the source sentence may have several translations in training data, and SMT system chooses the translation which has the highest probability score. Therefore, the translation which has more occurrences is more likely to be chosen. Differently, our method treats each translation equally and judge them from semantic perspective. In order to obtain all translation candidates for the terminology, we search the phrase table. The size of phrase table is usually very large, so we do hash operation on the phrase table and query terminology to improve efficiency. Then we obtain all terminology translations and filter some meaningless items.

<sup>2</sup> <http://www.statmt.org/moses>.

- Back translation. A back translation can be defined as the translation of a target sentence back to the original source language. In order to ensure the quality of the back translation, we call Youdao Translate API<sup>3</sup> interface instead of the reversed translation system constructed by ourselves. The input of the API is the text to be translated. In our case, it's a sentence which is the translation of the test sentence. The results returned from the API is the xml data structure.
- Selecting the best translation. For a test sentence, we have obtained several pseudo similar sentences. What we should do is to select the most similar sentence semantically and syntactically. We will detail this in the next section.

### 3.2 Compare Back Translation with Original Text

In this section, we will introduce three metrics to compare back translation with the original text. We think that terminology translation is more reliable when the similarity is higher between the back translation and the original sentence.

- Tree edit distance. Trees are among the most common and well-studied combinatorial structures in computer science. An optimal edit script between two trees is an edit script between them of minimum cost and this cost is the tree edit distance [12]. A tree edit model can be used to identify whether two sentences convey essentially the same meaning. In this paper, we use [13]'s method to calculate the tree edit distance between the dependency trees of two sentences. The smaller the distance, the greater the similarity of two sentences. We obtain dependency trees of sentences by Stanford NLP toolkit<sup>4</sup>. We assume that we will get a bad translation when the source sentence includes an inappropriate terminology in it, even the dependency structure of the translation will be different from the original sentence.
- Sentence semantic similarity. Sentences that share semantic and syntactic properties are thus mapped to similar vector representations [14]. In [14]'s work, they propose a model called skip-thought vectors which encode a sentence to predict the sentences around it. The results of experiments on the SemEval 2014 Task 1 show that skip-thought vectors learn representations that are well suited for semantic relatedness. Sentence similarity refers to the matching extent in semantics of two sentences which is a real number, the greater the value, the greater the similarity of the two sentences. We use the cosine similarity here.
- Language model perplexity. [10] use language model perplexity feature to estimate the quality of machine translation at sentence level. Inspired by them, we use this metric to measure the quality of back translation.

### 3.3 Corpus Acquisition

To perform our method, we need the test set which consists of sentences and the terminology in each sentence should be marked.

---

<sup>3</sup> <http://fanyi.youdao.com/openapi?path=data-mode>.

<sup>4</sup> <https://nlp.stanford.edu/software/nndep.shtml>.

We find that journals on the web are good resources, we just need to click on the title of the paper with no downloading and then we can obtain keywords and abstracts both in Chinese and English. We crawl the keywords and abstracts by using `urllib`<sup>5</sup> which is a python package that collects several modules for working with URLs. On the basis, we use another python package `BeautifulSoup`<sup>6</sup> to extract keywords and abstracts from the structured source files of the crawled web pages.

The next step is to obtain the sentences which the keywords are in. We detect sentence boundaries on English abstracts by using `OpenNLP`<sup>7</sup> which is a machine learning based toolkit for the processing of natural language text. For Chinese abstracts, we write rules to detect sentence boundaries. We use a rough but simple way to extract parallel sentences which the keywords are in. Each article has about four keywords, for each keyword, we locate the sentence containing this keyword in the Chinese abstract, and then check the corresponding index sentence in English abstract with extending two sentences window at most. This is because English abstract is not translated by Chinese abstract sentence by sentence in many articles. Besides, we make all English keywords and abstracts lowercase to avoid case matching problems.

## 4 Experiments

We conduct a pilot study for verifying whether back translation based strategy is useful for the identification and correction of terminology translation errors in the SMT system outputs.

### 4.1 Setup

Our training data consists of 16M mix-domain sentence pairs extracted from web by [15]’s acquisition method. We randomly choose 2k sentences as tuning [16] set from CWMTO9. The test set consists of 1657 sentences in English from the abstracts of a computer science’s journal. We collect 11, 224 bilingual terms from the keywords of the journal.

The word alignments were obtained by running `fast-align` [17] on the corpora in both directions and using the “grow-diag-final-and” balance strategy [18]. We adopted KEN Language Modeling Toolkit [19] to train a 5-gram language model with modified Kneser-Ney smoothing on the Xinhua portion of the Chinese<sup>8</sup>/English<sup>9</sup> Gigaword corpus.

We use [13]’s method to calculate the tree edit distance between dependency trees of two sentences. We obtain dependency trees of sentences by Stanford NLP toolkit.

---

<sup>5</sup> <https://docs.python.org/3/library/urllib.html>.

<sup>6</sup> <https://www.crummy.com/software/BeautifulSoup/>.

<sup>7</sup> <http://opennlp.apache.org/>.

<sup>8</sup> LDC2003T09 Gigaword Chinese Text Corpus Second Edition.

<sup>9</sup> LDC2009T13 Xinhua News Portion of English Gigaword Second Edition.

While the traditional sentence representation using mean pooled Word2Vec discards word order, SkipThoughts use a Recurrent Neural Network to capture the underlying sentence semantics. We use the pretrained model by [14] to compute a 4800 dimensional sentence representation.

We build several translation systems as follows:

- **Baseline:** We use Moses to construct English to Chinese translation system as our baseline system. The features used in baseline system include: (1) four translation probability features; (2) one language model feature; (3) distance-based and lexicalized distortion model feature; (4) word penalty; (5) phrase penalty.
- **Baseline+BiTerm:** [20] prove that concatenating the training data and the terms perform better than more complex techniques. We take the bilingual terms as parallel sentence pairs and add them into the training corpus.
- **Baseline+BTTC:** Performing our method on the outputs of the Baseline system.
- **Baseline+BiTerm+BTTC:** Performing our method on the outputs of the Baseline+BiTerm system.

For the original terminology translation in the SMT system outputs, we think it may be wrong if it satisfies the following two conditions at the same time: (1) the result of the highest language model perplexity minus the original terminology translation's perplexity score is greater than the threshold value which we empirically set as 0.015; (2) its semantic similarity is lower than the highest score.

As for translation suggestion, we use three methods: (1) selecting the translation candidate whose back translation is the most similar to the test sentence semantically; (2) selecting the translation candidate whose back translation has the lowest tree-edit distance; (3) selecting the translation candidate whose back translation has the maximum difference between semantic similarity and tree-edit distance.

## 4.2 Evaluation Metrics

We conduct our method on the test set, with the aim to verify whether back translation based terminology-check method is able to identify the wrongly translated terminology and suggest a better translation. The basic evaluation metric is the precision rate (PR). Precision rate is defined as the percentage of the terms that are correctly translated as follow:

$$PR = \frac{\# \text{ of correctly translated terms}}{\text{Total } \# \text{ of terms}} \quad (2)$$

## 5 Results

Table 3 gives our experiment results. From this table, we can see that three suggestion methods all have positive effects, and semantic similarity method works better than the tree-edit distance method. For Baseline system, the tree-edit method achieves 0.36% precision improvement and the semantic method achieves 0.42% precision improvement. Baseline+BiTerm system also gives an evidence of this, the tree-edit method



achieves 1.09% precision improvement and the semantic method achieves 1.21% precision improvement. Combing two metrics works best, which achieves 0.48% and 1.51% precision improvement on two systems respectively. The results also show that the BTTC can work better on the strong translation system. This is mainly because the strong translation system is trained from the higher quality corpora which contains more useful translation information. Therefore, our method is more likely to retrieve the correct terminology translation and make corrections.

**Table 3.** Performance of BTTC on different systems

Methods	Precision (%)
Baseline	25.05
Baseline+BTTC (tree-edit)	25.41
Baseline+BTTC (semantic)	25.47
Baseline+BTTC (semantic + tree-edit)	25.53
Baseline+BiTerm	54.19
Baseline+BiTerm+BTTC (tree-edit)	55.28
Baseline+BiTerm+BTTC (semantic)	55.40
Baseline+BiTerm+BTTC (semantic+tree-edit)	55.70

In order to know in what respects our method improve performance of translation, we manually analyze some test sentences and give some examples in Table 4. The back translations of all three sentences' original translations are semantically deviated from the source sentences. However, the replaced translation with the right terminology translation is more contextual consistent and their back translation is semantically similar to the source sentences.

**Table 4.** Translation examples

**Src:** *therefore the problem of quantum grammars generating quantum regular languages is solved.*

**Baseline+BiTerm:** 经常语言

**Biseline+Biterm+BTTC:** 正规语言

**Src:** *in order to optimize data locality and communication overhead, this paper proposes a novel alternate tiling stencil algorithm on distributed memory machines by exploiting the property of the iterative algorithm.*

**Baseline+BiTerm:** 数据区域性

**Biseline+Biterm+BTTC:** 数据局部性

We find that although many wrongly translated terminologies are corrected by BTTC, but the overall performance is not obvious. The reason is that some correct terminology translations are wrongly revised by BTTC. Considering a scenario where the user is dissatisfied with the outputs of the translation system, more specifically, he or she think the terminology translation is wrong. In such case, we get the feedback and know which terminology need to be corrected. Table 5 shows the better performance of our method in such situation. We perform our post-editing method on those true mistakes. The results show that BTTC achieves 0.96% and 3.38% precision improvement on Baseline system and Baseline+BiTerm system respectively.

**Table 5.** Performance of BTTC on true mistakes

Methods	Precision (%)
Baseline+BTTC	26.01
Baseline+BiTerm+BTTC	57.57

In addition, we find the sentence vector causes some mistakes. Table 6 shows an example. Obviously, the True\_backtran is more similar with the Gold sentence, but the semantic similarity of True\_backtran is 0.848 and lower than False\_backtran's score, which is 0.972.

**Table 6.** Inappropriate scored examples

**Gold:** *The bus network is more suitable when you want to connect to some computers.*

**True:** *当你想连接一些电脑的时候，总线网络更合适。*

**True\_backtran:** *When you want to connect to some computers, the bus network is more appropriate.*

**False:** *当你想连接一些电脑的时候，公交网络更合适。*

**False\_backtran:** *When you want to connect to some computer time, public transportation network more suitable.*

## 6 Conclusion and Future Works

We propose a back translation based method to automatically identify terminology translation errors in the SMT system outputs and suggest a better translation. Our method relies on an external generic reversed MT engine and needs to know which is the terminology in the test sentence. We propose three metrics to measure the quality of back translation. Experimental results show that our method can suggest better terminology translations for both weak and strong translation systems. The performance of our method is better when the training data contains more translation information such as domain terminology. Besides, the performance can be further improved when the identification precision improves.

However, the strategies of measuring back translation are roughly simple and coarse in this paper. Complicated approach should be taken into account during identifying the true mistakes. In future work, we also consider representing the semantic of a sentence more accurately. In addition, acquiring terminology dictionary is also meaningful for our work, and each item in the dictionary corresponds to many possible translations.

**Acknowledgments.** This research work is supported by National Natural Science Foundation of China (Grants No. 61373097, No. 61672367, No. 61672368, No. 61331011), the Research Foundation of the Ministry of Education and China Mobile, MCM20150602 and the Science and Technology Plan of Jiangsu, SBK2015022101. The authors would like to thank the anonymous reviewers for their insightful comments and suggestions.

## References

1. Axelrod, A., He, X., Gao, J.: Domain adaptation via pseudo in-domain data selection. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 355–362. Association for Computational Linguistics, Edinburgh (2011)
2. Carl, M., Langlais, P.: An intelligent terminology database as a pre-processor for statistical machine translation. In: Second International Workshop on Computational Terminology COLING-02 on COMPUTERM 2002, vol. 14, pp. 1–7. Association for Computational Linguistics (2002)
3. Skadiņš, R., Pinnis, M., Gornostay, T., Vasiljevs, A.: Application of online terminology services in statistical machine translation. In: Proceedings of the XIV Machine Translation Summit, MT Summit XIV, France, pp. 281–286 (2013)
4. Meng, F., Xiong, D., Jiang, W., Liu, Q.: Modeling term translation for document-informed machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 546–556. Association for Computational Linguistics, Doha (2014)
5. Pinnis, M., Skadiņš, R.: MT adaptation for underresourced domains—what works and what not. In: Proceedings of the 5th International Conference Baltic HLT, p. 176. IOS Press (2012)
6. Ren, Z., Lu, Y., Cao, J., Liu, Q., Huang, Y.: Improving statistical machine translation using domain bilingual multiword expressions. In: Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications, pp. 47–54. Association for Computational Linguistics, Suntec (2009)
7. Itagaki, M., Aikawa, T.: Post-MT term swapper: supplementing a statistical machine translation system with a user dictionary. In: Proceedings of the 6th International Conference on Language Resources and Evaluation. European Language Resources Association, Marrakech (2008)
8. Bosca, A., Nikoulina, V., Dymetman, M.: A lightweight terminology verification service for external machine translation engines. In: Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 49–52. Association for Computational Linguistics, Gothenburg (2014)
9. Xiong, D., Zhang, M., Li, H.: Error detection for statistical machine translation using linguistic features. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 604–611. Association for Computational Linguistics, Uppsala (2010)

10. Wisniewski, G., Pécheux, N., Allauzen, A.: LIMS submission for WMT'14 QE task. In: Proceedings of the 9th Workshop on Statistical Machine Translation, pp. 348–354. Association for Computational Linguistics, Baltimore (2014)
11. José, G.C., de Souza, J.G.-R., Buck, C., Turchi, M., Negri, M.: FBK-UPV-UEdin participation in the WMT14 quality estimation shared-task. In: Proceedings of the 9th Workshop on Statistical Machine Translation, pp. 322–328. Association for Computational Linguistics, Baltimore (2014)
12. Bille, P.: A survey on tree edit distance and related problems. *Theoret. Comput. Sci.* **337**(1), 217–239 (2005)
13. Yao, X., Van Durme, B., Callison-Burch, C., Clark, P.: Answer extraction as sequence tagging with tree edit distance. In: Proceedings of North American Chapter of the Association for Computational Linguistics, pp. 9–14. Association for Computational Linguistics Atlanta (2013)
14. Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in Neural Information Processing Systems, pp. 3294–3302 (2015)
15. Liu, L., Hong, Y., Lu, J., Lang, J., Ji, H., Yao, J.M.: An iterative link-based method for parallel web page mining. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1216–1224. Association for Computational Linguistics, Doha (2014)
16. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 160–167. Association for Computational Linguistics, Sapporo (2003)
17. Dyer, C., Chahuneau, V., Smith, N.A.: A simple, fast, and effective reparameterization of IBM model 2. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 644–649. Association for Computational Linguistics, Atlanta (2013)
18. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 48–54. Association for Computational Linguistics, Edmonton (2003)
19. Heafield, K., Pouzyrevsky, I., Clark, J.H., Koehn, P.: Scalable modified Kneser-Ney language model estimation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 690–696. Association for Computational Linguistics, Sofia (2013)
20. Bouamor, D., Semmar, N., Zweigenbaum, P.: Identifying bilingual multi-word expressions for statistical machine translation. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, pp. 674–679. European Language Resources Association, Istanbul (2012)