Fast and Accurate Bilingual Lexicon Induction via Matching Optimization

Zewen Chi[1,2,3], Heyan Huang*[1], Shenjian Zhao[4],
Heng-Da Xu[1], and Xian-Ling Mao[1]

[1] Department of Computer Science and Technology,
Beijing Institute of Technology, China
[2] CETC Big Data Research Institute Co., Ltd., Guiyang 550022
[3] Big Data Application on lmproving Government Governance Capabilities National Engineering Laboratory Guiyang 550022
[4] ByteDance Inc.
czwin32768@gmail.com hhy63@bit.edu.cn
zhaoshenjian.01@bytedance.com dadamrxx@gmail.com
maoxl@bit.edu.cn

Abstract. Most recent state-of-the-art approaches are proposed to utilize the pre-trained word embeddings for bilingual lexicon induction. However, the word embeddings introduce noises for both frequent and rare words. Especially in the case of rare words, embeddings of which are always not well learned due to their low occurrence in the training data. In order to alleviate the above problem, we propose BLIMO, a simple yet effective approach for automatic lexicon induction. It does not introduce word embeddings but converts the lexicon induction problem into a maximum weighted matching problem, which could be efficiently solved by the matching optimization with greedy search. Empirical experiments further demonstrate that our proposed method outperforms state-of-the-arts baselines greatly on two standard benchmarks.

1 Introduction

Bilingual lexicons are crucial for cross language processing, since they boost the performance of downstream tasks such as multilingual classification and machine translation [10, 15, 7]. However, high quality bilingual lexicons are not always available, especially for low resource languages. Additionally, most bilingual lexicons only cover frequent words while a large amount of rare words are missing. With new words emerging, more words in the long tail are absent from these lexicons. Thus, automatically inducing lexicons with moderate supervision is essential to extend the standard lexicons.

Most previous lexicon induction methods work in a soft matching way — for each target word, they only give a list of candidate words with corresponding probabilities. The induced lexicons can be noisy in real applications. Recently,

* Corresponding author

the most popular lexicon induction method is the bilingual word embedding transforming, which maps the bilingual word embeddings into one space with a transforming matrix. Specifically, they learn a linear transformation from the source embedding space to the target embedding space, based on the assumption that word embeddings of different languages have similar geometric arrangements in there corresponding embeddings spaces. After that, the learned transforming matrix acts like a soft alignment between the bilingual words, which achieves very impressive results on lexical induction tasks [24, 21, 7, 4, 18].

However, we argue that the current state-of-the-art methods by transforming word embeddings are not necessarily the best for automatic lexicon induction, because they highly depend on the quality of pre-trained word embeddings. Because words with similar meanings tends to have similar word vectors, and in the setting of embeddings transforming, source words are likely to be mistakenly aligned to target words with similar embeddings. Furthermore, we also find that rare words are not well aligned by embedding transforming methods. Word embeddings of rare words are always poorly learned since they does not appear enough times in the data for the embedding training. Due to the power-law distribution, the 2% of the most frequent words could take 98% of the total training data, which results in the relatively low quantity of rare word embeddings. In such case, word embeddings will be severely noisy for lexicon induction on rare words. In the meantime, bilingual lexicons of rare words are usually more crucial than some frequent words in the downstream tasks. For example, rare words in machine translation are prone to be some informed name entities or even unknown words (UNK), obtaining the lexicons of these rare words may significantly improve the performance of machine translation.

In this paper, we propose BLIMO (short for Bilingual Lexicon Induction via Matching Optimization), a fast yet accurate approach for bilingual lexicon induction, which abandons the soft matching approach and does not introduce noisy word embeddings for the lexicon induction process. Following previous work, we propose to exploit the easily acquired bilingual parallel data, maximizing the similarity of source and target sentence representations. The sentence representation is the normalized summation of the word representations, but different to previous work, we use the one-hot vector as the word representations, which does not have the above mentioned problems of word embeddings. Specifically, we reduce the lexicon induction problem to maximum weighted matching in a bipartite graph. By assuming the property of *lexicon bijection* (see Section 3.3), which is quite reasonably for rare words, lexicon induction in our scenario could be further modeled as a matching optimization problem. For efficiency, we propose a greedy algorithm to find the approximated solution of the matching optimization, which is very fast and giving very impressive results in practice.

In the experiments, we conduct experiments on English-Italian and Japanese-English benchmarks. Our proposed BLIMO gives better results than existing methods greatly on both benchmarks. To the best of our knowledge, we achieve the best reported results on English-Italian data, which boosts the state-of-the-art performance [7] of lexicon induction from 66.2% to 74.1%, obtaining a

significant improvement of 8 absolute percent on this standard benchmark. Our proposed method outperforms state-of-the-art baselines on both frequent words and rare words, and the accuracy gap on rare words are larger than on frequent words. This shows the advantages of our proposed method by abandoning word embeddings. Additionally, our BLIMO works very fast practically, which can extract 80K English-Italian lexicons within 10 minutes on 2M parallel sentences.

2 Related Work

Methods for bilingual lexicon induction can be classified into three categories, i.e., supervised methods, weakly supervised and unsupervised methods. Supervised methods mostly exploit a bilingual lexicon or parallel corpus to model the relationship of words between different languages. The weakly supervised methods could use a small number of seed lexicons, while unsupervised methods only make use of monolingual corpus.

2.1 Supervised Methods

In the early research of lexicon induction, most related works focus on word alignment problem in machine translation, which aims to find the word alignment of a bilingual sentence-aligned corpus with language-independent statistical methods [17]. These can be viewed as the earliest works on the bilingual lexicon extraction tasks. They exploit similarity functions to align similar words or use some other statistical methods like hidden Markov models [20, 11, 14, 5]. These works pay more attention to local alignment of words between sentences rather than obtaining global lexicons, and the lexicon induction by unsupervised word alignment may need many iterations with the EM algorithm, which is very time consuming.

In recent years, most approaches are based on bilingual embedding mappings. Mikolov et. al. (2013) [15] first use word embeddings in the extraction of lexicons. Supervised by a seed lexicon, their method learns a linear transformation matrix to minimize squared the Euclidean distance between transformed source word vectors and target word vectors. Word translation is extracted by searching nearest neighbors. Following works [9, 13, 23, 2] adopt similar idea but they additionally apply a canonical correlation analysis or add an orthogonality constraint to the mapping matrix, which gain a performance improvement.

The method proposed by AP et. al. (2014) [1] learns to reconstruct bag-of-words representations of aligned sentences without using word alignment or seed lexicons. While recent work by Smith et. al. (2017) [21] exploits parallel corpus to learn a transformation matrix. They define a vector representation of sentence by a normalized sum over the word vectors, and view the parallel corpus as a dictionary of "average word" pairs. With these "word" pairs, they construct a "pseudo-dictionary" as the seed dictionary to learn a orthogonal transformation in embedding spaces. However, the above two works still rely on the word embeddings for word representation.

2.2 Weakly supervised and unsupervised methods

Recent works show that weakly supervised and unsupervised methods can also obtain good performances on the bilingual lexicon induction. Artetxe et. al. (2017) [3] propose a self-learning method that separates the task into a dictionary extraction step and a embedding mapping step, and then iterates these two steps with a seed dictionary, which contains only 25 word pairs. Artetxe et. al. (2018) [4] further extend this two-step framework with a fully unsupervised initialization based on a simple assumption that the embedding spaces are perfectly isometric, and similarity matrices of monolingual word embeddings should be equivalent up to a permutation of their rows and columns. While other unsupervised methods employ adversarial training, they learn a discriminator and a mapping matrix, in which the discriminator is trained to determine whether an word embedding comes from source or target languages, while the mapping matrix is trained to fool the discriminator through transforming source word embeddings distribution close to target word embeddings distribution [25, 7].

These approaches differ from ours in following aspects. They all aim to learn a cross-lingual word representation and then learn a cross-lingual classifier or extract lexicons with these representations. However, our approach views the bilingual lexicon induction as a deciphering task and directly learns the bilingual dictionary. Besides, most of these methods all relies on the distributed representation of words. These sentences are represented as the sum or average of the distributed representation of words, which causes information loss especially for long sentences.

3 Approach

3.1 BLIMO

In this section, we will describe our proposed BLIMO in detail. Suppose we have n parallel sentences, denoted as $\{S_i,\ T_i\}_{i=1}^{n}$. S_i is the i-th source sentence consists of words $\{W^s_{S_{i,1}}, W^s_{S_{i,2}}, \ldots, W^s_{S_{i,\mathrm{len}(S_i)}}\}$, and $T_i = \{W^t_{T_{i,1}}, W^t_{T_{i,2}}, \ldots, W^t_{T_{i,\mathrm{len}(T_i)}}\}$ is the target sentence corresponding to S_i. W^s_* and W^t_* are words in source language and target language respectively. 'len(S_i)' and 'len(T_i)' is the number of words in the source sentence and the target sentence. For a clearer illustration, we map each word into a v-dimensional one hot vector $\mathbf{h}(\cdot)$. To be general, suppose we have an embedding matrix $\mathbf{E}_s \in \mathbb{R}^{m\times v}$ for the source language. The representation of each source sentence could be sum of word vectors, e.g.,

$$\mathbf{s}_i = \sum_{j=1}^{\mathrm{len}(S_i)} \mathbf{E}_s \mathbf{h}(W^s_{S_{i,j}}) = \mathbf{E}_s \sum_{j=1}^{\mathrm{len}(S_i)} \mathbf{h}(W^s_{S_{i,j}}), \tag{1}$$

$\mathbf{E}_s$ could be a distributed embedding matrix. It should be noted that we only use the embedding $\mathbf{E}_s$ in deduction, and it will be eliminated in the following steps.

Given a sentence pair $\langle S_i,\ T_i\rangle$, we measure the distance of these two sentences by,

$$\mathrm{dist}(S_i, T_i) = -\mathrm{norm}(\mathbf{s}_i)^\intercal \mathrm{norm}(\mathbf{t}_i), \tag{2}$$

in which $\mathrm{norm}(\cdot)$ is a function that normalize sentence vectors. We need to find a mapping between source words and target words that minimize the $\mathrm{dist}(\cdot,\cdot)$ for the corpus. Specially, we use a mapping function $p(\cdot)$ to map the target words $\{W^t_1, W^t_2, \ldots, W^t_v\}$ to source words $\{W^s_{k_1}, W^s_{k_2}, \ldots, W^s_{k_v}\}$. Then $\mathbf{t}_i$ could be written as,

$$\mathbf{t}_i = \sum_{j=1}^{\mathrm{len}(T_i)} \mathbf{E}_s \mathbf{h}(p(W^t_{T_{i,j}})). \tag{3}$$

To make the problem tractable, we further assume that each word in the source language can be mapped to only one word in the target language (See the following section for detail). Then $p(\cdot)$ could be written as a row transformation matrix $\mathbf{D}$ of dimension $n \times n$, that,

$$\mathbf{h}(p(W^t_{T_{i,j}})) = \mathbf{D}\mathbf{h}(W^t_{T_{i,j}}). \tag{4}$$

With the help of mapping matrix $\mathbf{D}$, Eq. (3) could be written as

$$\mathbf{t}_i = \sum_{j=1}^{\mathrm{len}(T_i)} \mathbf{E}_s \mathbf{D}\mathbf{h}(W^t_{T_{i,j}}) = \mathbf{E}_s \mathbf{D} \sum_{j=1}^{\mathrm{len}(T_i)} \mathbf{h}(W^t_{T_{i,j}}). \tag{5}$$

The distance of source corpus and target corpus is

$$\mathrm{dist}(S, T) = \sum_{i=1}^{n} \mathrm{dist}(S_i, T_i) = -\sum_{i=1}^{n} \mathrm{norm}(\mathbf{s}_i)^\intercal \mathrm{norm}(\mathbf{t}_i). \tag{6}$$

For computation efficiency, we use L^2-norm as the normalization function. In this setting, we set $\mathbf{E}_s$ to the identity matrix because of the sparsity of $\sum_{j=1}^{\mathrm{len}(S_i)} \mathbf{h}(W^s_{S_{i,j}})$. Thus, $\mathrm{dist}(S, T)$ could be further simplified as following,

$$\mathrm{dist}(S, T) = -\sum_{i=1}^{n} \frac{\mathbf{s}^t_i}{\sqrt{\mathbf{s}^\intercal_i \mathbf{s}_i}} \frac{\mathbf{t}_i}{\sqrt{\mathbf{t}^\intercal_i \mathbf{t}_i}},$$

in which

$$\begin{aligned} \frac{\mathbf{s}_i}{\sqrt{\mathbf{s}^\intercal_i \mathbf{s}_i}} &= \frac{\sum_{j=1}^{\mathrm{len}(S_i)} \mathbf{h}(W^s_{S_{i,j}})}{\sqrt{(\sum_{j=1}^{\mathrm{len}(S_i)} \mathbf{h}(W^s_{S_{i,j}})^\intercal)(\sum_{j=1}^{\mathrm{len}(S_i)} \mathbf{h}(W^s_{S_{i,j}}))}} \\ &= \mathrm{norm}(\mathbf{h}_{S_i}) \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{\mathbf{t}_i}{\sqrt{\mathbf{t}^\intercal_i \mathbf{t}_i}} &= \frac{\mathbf{D}\sum_{j=1}^{\mathrm{len}(T_i)} \mathbf{h}(W^t_{T_{i,j}})}{\sqrt{(\sum_{j=1}^{\mathrm{len}(T_i)} \mathbf{h}(W^t_{T_{i,j}})^\intercal)\mathbf{D}^\intercal\mathbf{D}(\sum_{j=1}^{\mathrm{len}(T_i)} \mathbf{h}(W^t_{T_{i,j}}))}} \\ &= \mathbf{D}\ \mathrm{norm}(\mathbf{h}_{T_i}) \end{aligned} \tag{8}$$

We combine representation of all normalized sentence vectors into a single matrix for simplification. For instance, $\mathbf{S} = [\text{norm}(\mathbf{h}_{S_1}), \text{norm}(\mathbf{h}_{S_2}), \ldots, \text{norm}(\mathbf{h}_{S_n})]$ and $\mathbf{T} = [\text{norm}(\mathbf{h}_{T_1}), \text{norm}(\mathbf{h}_{T_2}), \ldots, \text{norm}(\mathbf{h}_{T_n})]$ is the representation of source corpus and target corpus, respectively. In this setting, the only variable we need solve is the mapping $\mathbf{D}$ between two languages. Therefore, the objective is to find a mapping matrix $\mathbf{D}$ such that the distance between parallel corpus is minimized:

$$\begin{aligned} \arg\min_{\mathbf{D}} \text{dist}(S, T) &= \arg\min_{\mathbf{D}} \sum_{i=1}^{n} \text{dist}(S_i, T_i) \\ &= \arg\max_{\mathbf{D}} \text{tr}\Big(\mathbf{T}\mathbf{S}^\intercal\mathbf{D}\Big) \end{aligned} \tag{9}$$

where $\text{tr}(\cdot)$ is the trace operation (the sum of the entries in the main diagonal of the matrix). Let $\mathbf{A} := \mathbf{T}\mathbf{S}^\intercal$. With the lexicon bijection assumption (see Section 3.3), each word in the source language can be mapped to only one word in the target language, in which case, $\mathbf{D}$ is permutation matrix. We can optimize the objective function by finding a permutation of $\mathbf{A}$'s columns. Such optimization problem can be reduced to the problem of finding a maximum weighted matching in a bipartite graph where $\mathbf{A}_{ij}$ is the weight of the edge connecting i-th vertex on the left side and j-th vertex on the right side.

3.2 Why One-Hot Word Representation

To give a further explanation of why we should use one-hot vector as the word representation, we illustrate the reason in following two aspects: (1)We find that currently popular distributed representation of words may introduce noisy and leads to bad performances of similar words and rare words on lexicon induction. (2) As it is mentioned above, we need to solve maximum weighted matching problem on a bipartite graph. If we can limit the weight to positive and make the weight matrix $\mathbf{A}$ very sparse, then we can save a lot of memory space and computational resource. We find that using one-hot vectors can exactly satisfy these two conditions.

3.3 Lexicon Bijection Assumption

In this section, we introduce a lexicon bijection assumption in the modeling of lexicon induction, which means the words in the source and target languages should be a one-to-one mapping. Although the lexicon permutation assumption is a really strong assumption, it still makes sense because we mainly focus on boosting the lexicon induction performance rare words, which are almost bijective. Empirically, our assumption does not harm the accuracy of lexicon induction and the experiments show that our proposed method gives really good results on rare words.

3.4 Matching Optimization

According to the previous description, we want to find a matching of **A**'s columns to maximize its trace, which is a maximum weighted matching problem or a linear sum assignment problem (LSAP). There are a large number of algorithms have been developed for LSAP and the best sequential algorithms for the LSAP requires a time complexity of $O(v^3)$ in the worst-case, where n is the size of the problem [6]. However, when it comes to a larger vocabulary, it's not applicable to extract the lexicon with a complexity of $O(v^3)$. So we adopt an alternative method to solve this problem: we iteratively select the highest-weight item (i, j) in A and remove the corresponding row and column until all the words are aligned. This procedure have a time complexity of $O(v^2 \log(v))$ in the worst case. Due to the sparsity of the matrix **A**, we can only sort those non-zero values and actually the expected time complexity is $O(Cv^2 \log(v))$, where C is a constant represents the density of the matrix. There is no iterative steps in our proposed method, and empirically our method always gives accurate bilingual lexicons with really fast speed. For example, we can extract 80K English-Italian lexicons within 10 minutes on 2M parallel sentences.

4 Experiments

In this section, we first evaluate our proposed method on standard benchmarks of lexicon inductions, and then make a comparison with a variety of currently state-of-the-art baselines. Moreover, we apply our learned bilingual lexicon in a state-of-the-art machine translation system, trying to verifying that whether the induced bilingual lexicon can help to boost the performance of a modern machine translation system.

4.1 Experiments on Bilingual Lexicon Extraction

Experiment setup This task aims to find the translations in target language with the given words in source language. We evaluate our approach on standard lexicon induction benchmarks, the English-Italian and the Japanese-English datasets, respectively. The English-Italian dataset is provided by Dinu et. al. (2014) [8] . Specifically, the English-Italian test set contains 1500 words. These words are divided into five frequency-sorted bins (1-5k, 5-20k, 20-50k, 50-100k and 100-200k), and each bin contains 300 words. To give a fair comparison with previous supervised approaches, in the English-Italian task we use the same parallel corpus as used in [21] , which is a 2M English-Italian parallel corpus from the Europarl corpus [12].

English and Italian are similar to each other, because they belong to the same Indo-European language family. In order to show more strengths of our method, we conduct the experiments on Japanese-English language pair, which are two very different languages, belonging to the Japanese-Ryukyuan language family and Indo-European language family, respectively. We use the ASPEC

dataset [16] for training, which is a corpus from the scientific paper domain. As for the Japanese-English test set, we cut the first 6500 words of the full dataset [7] as the test set and split them into 13 bins according to their frequency rank.

We set the most frequent 80K words as the vocabulary in both tasks. Both tasks take the polysemy of words into account, which helps us to have a more accurate evaluation of the bilingual lexicons quality. This setup enables us to detect the performance of methods on words with different frequencies, so that we can evaluate our method from another perspectives.

Word ranking by frequency	Mikolov et al.[15]	Dinu et al.[8]	CCA[21]	Smith et al.[21]	Artetxe et al. [4]	Conneau et al.[7]	This work
0-5k	0.607	0.650	0.633	0.690	-	-	**0.807**
5-20k	0.463	0.540	0.477	0.610	-	-	**0.800**
20-50k	0.280	0.350	0.343	0.403	-	-	**0.787**
50-100k	0.193	0.217	0.190	0.253	-	-	**0.670**
100-200k	0.147	0.163	0.163	0.200	-	-	**0.640**
average	0.338	0.385	0.361	0.431	0.481	0.662	**0.741**

Table 1: Translation precision @1 from English to Italian with different word frequency. Results are obtained from [4, 7, 21].

Quantitative Results Our results on the English-Italian test set are reported in Table 1, as well as the results of Mikolov, Faruqui, Dinu and Smith reported in [21] , [4] and [7] . We make a comparison with these six different methods, including linear transformation learning method presented by Mikolov, Faruqui's method using Scikit-learn's implementation of CCA and Smith's method supervised by parallel corpus, as well as Artetxe's two-step method and Conneau's adversarial training method.

As shown in Table 1, our method achieves a remarkably high precision on both common words and rare words, which could support our motivation mentioned in the Introduction. It is worth mentioning that comparing to previous works, the performance of our method does not have a big drop off for rare words. Most of previous works give really bad results on rare words such as the words of 50-100k and 100-200k, ranked by frequency. Especially for the words ranked as 100-200k, most results reported by previous works are around 20%, which is significantly lower than ours (64%). This shows our proposed method is really superior to baselines on rare words. Moreover, our proposed also work better on frequent words than previous work, and finally our proposed method achieve an accuracy of 74.1% on all words, which is significantly better than baselines.

Our results on the Japanese-English are shown in Figure 1. In this evaluation, we compare our method with [21] and [7] , which are the most representative methods in unsupervised methods and supervised methods, respectively. We

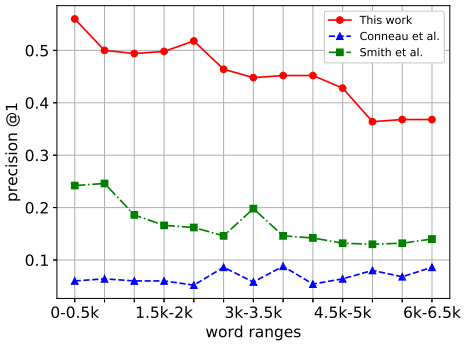

Fig. 1: Word translation precision@1 from Japanese to English with different word frequency.

evaluate the method of [7] [5] and [21] [6] based on their implementation. The word embeddings used in these two evaluation are pre-trained fastText Wikipedia embeddings[7].

We show the results in Figure 1, and we find that on the Japanese-English dataset, our method also outperforms baselines with a relatively large margin. Specifically, our method achieves an accuracy of 45.5%, which is better than compared models of [7] and [21] , 16.7% and 7%, respectively. Besides, the running time of their method is 9 hours (CPU time) and 30 minutes (GPU time), respectively. Our method only takes 10 minutes (CPU time) to extract the 80K Japanese-English lexicon.

Qualitative Analysis We provide some words in the test set and their translations predicted by various model in Table 2. To investigate the behavior of each approach, we arrange these words according to their frequency. In Table 2, we could find that the unsupervised method [7] fail to learn a reasonable mapping. Japanese and English may be too different for unsupervised method to learn the mapping successfully. Both supervised methods are able to learn good mapping on some common words, such as 学校 and 観光. アメリカ and 石油 are not correctly mapped because there are multiple translations for these words. For instance, 北米 and 油 are mapped 'america' and 'oil' in our approach. The method of [21] fail to generate a good translation for rare words and similar words, for example, 坂本 and マニラ.

[5] https://github.com/facebookresearch/MUSE
[6] https://github.com/Babylonpartners/fastText_multilingual
[7] https://github.com/facebookresearch/fastText

Japanese	Groundtruth English	Conneau et al.	Smith et al.	Ours
学校	[schools, school]	choreutidae	schools	school
アメリカ	[america]	indian	united	american
観光	[tourist, sightseeing]	hepialidae	tourist	sightseeing
石油	[oil]	bucculatricidae	petroleum	petroleum
坂本	[sakamoto]	cosmopterigidae	mrged	sakamoto
ハワイ	[hawaii]	sulawesi	island	hawaii
注釈	[annotated, annotation]	annotated	commentaries	annotation
時折	[occasionally]	moth	especially	occasionally
土器	[earthenware]	blastobasidae	excavated	earthenware
マニラ	[manila]	tarawa	boarded	manila

Table 2: Word translation samples from the Japanese-to-English task.

Word alignment	Vocabulary	Share Embedding	Parameters	BLEU
Not aligned	80K words	False	277 M	26.80
Randomly aligned	80K words	True	199 M	26.70
Not aligned	40K subwords (BPE)	False	199 M	27.36
Aligned by lexicon	80K words	True	199 M	**27.75**

Table 3: The effect of words alignment for neural machine translation on ASPEC Japanese-to-English task.

4.2 Experiments on machine translation

Bilingual vocabulary could be used in many NLP tasks, such as neural machine translation. As explained in Section 4.1, building a bilingual vocabulary of English and Japanese is challenging. Our experiments on bilingual lexicon extraction further show that, not only the unsupervised way fails to build a satisfactory bilingual vocabulary, but also the supervised method using word embedding could not find a promising relation. In contrast, our method is able to align both common and rare Japanese words into English with relatively higher accuracy. In order to verify the effect of word alignment, we conduct comparison of various settings on neural machine translation tasks, which is evaluated using BLEU.

In contrast to [2] , we learn bilingual word mapping directly, instead of learning a transformation between embedding spaces. Thus, we train the neural machine translation in supervised mode and learn word embedding in the meantime. We use Transformer [22] as our baseline model. To ease the effect of overfitting, we set the hidden size of Transformer as 256.

We list the BLEU scores on the Japanese-to-English task in Table 3. All these models are trained for 100,000 steps. It can be observed that although all the four models are trained on the same dataset, we easily reach the best performance comparing to those not or randomly aligned models, which confirms the

effectiveness of our word alignment. Even more surprising, the model with word alignment performs better than the BPE method [19]. Note that the words are randomly aligned in the second model, but it still achieves a good performance. It suggests that even though our inducted lexicon is not perfect, it can still helps the NMT model to translate better.

5 Conclusions

In this paper, we propose BLIMO, a method that directly extract bilingual lexicon without using distributed representation of words. The experimental results on English-Italian and Japanese-English word translation task, as well as the Japanese-English machine translation task demonstrate that our method can extract high-quality bilingual lexicons from parallel corpus. For the future work, we would like to relax the bijection hypothesis of lexicon and also seek more reasonable approximation algorithms.

Acknowledgement

The work is supported by SFSMBRP(2018YFB1005100), BIGKE(No. 20160754021), NSFC (No. 61772076 and 61751201), NSFB (No. Z181100008918002), Major Project of Zhijiang Lab (No. 2019DH0ZX01), and CETC(No. w-2018018).

References

1. AP, S.C., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V.C., Saha, A.: An autoencoder approach to learning bilingual word representations. In: Advances in Neural Information Processing Systems. pp. 1853–1861 (2014)
2. Artetxe, M., Labaka, G., Agirre, E.: Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 2289–2294 (2016)
3. Artetxe, M., Labaka, G., Agirre, E.: Learning bilingual word embeddings with (almost) no bilingual data. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 451–462 (2017)
4. Artetxe, M., Labaka, G., Agirre, E.: A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. ACL (2018)
5. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational linguistics **19**(2), 263–311 (1993)
6. Burkard, R.E., Cela, E.: Linear assignment problems and extensions. In: Handbook of combinatorial optimization, pp. 75–149. Springer (1999)
7. Conneau, A., Lample, G., Ranzato, M., Denoyer, L., Jégou, H.: Word translation without parallel data. arXiv preprint arXiv:1710.04087 (2017)
8. Dinu, G., Lazaridou, A., Baroni, M.: Improving zero-shot learning by mitigating the hubness problem. Computer Science **9284**, 135–151 (2014)

9. Faruqui, M., Dyer, C.: Improving vector space word representations using multilingual correlation. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 462–471 (2014)
10. Gliozzo, A., Strapparava, C.: Exploiting comparable corpora and bilingual dictionaries for cross-language text categorization. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. pp. 553–560. Association for Computational Linguistics (2006)
11. Ker, S.J., Chang, J.S.: A class-based approach to word alignment. Computational linguistics **23**(2), 313–343 (1997)
12. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT summit. vol. 5, pp. 79–86 (2005)
13. Lu, A., Wang, W., Bansal, M., Gimpel, K., Livescu, K.: Deep multilingual correlation for improved word embeddings. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 250–256 (2015)
14. Melamed, I.D.: Models of translational equivalence among words. Computational Linguistics **26**(2), 221–249 (2000)
15. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. Computer Science (2013)
16. Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S., Isahara, H.: Aspec: Asian scientific paper excerpt corpus. In: LREC (2016)
17. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. MIT Press (2003)
18. Riley, P., Gildea, D.: Orthographic features for bilingual lexicon induction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 390–394 (2018)
19. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)
20. Smadja, F., McKeown, K.R., Hatzivassiloglou, V.: Translating collocations for bilingual lexicons: A statistical approach. Computational linguistics **22**(1), 1–38 (1996)
21. Smith, S.L., Turban, D.H., Hamblin, S., Hammerla, N.Y.: Offline bilingual word vectors, orthogonal transformations and the inverted softmax. ICLR (2017)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
23. Xing, C., Wang, D., Liu, C., Lin, Y.: Normalized word embedding and orthogonal transform for bilingual word translation. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1006–1011 (2015)
24. Zhang, M., Liu, Y., Luan, H.B., Sun, M., Izuha, T., Hao, J.: Building earth mover's distance on bilingual word embeddings for machine translation. In: AAAI. pp. 2870–2876 (2016)
25. Zhang, M., Liu, Y., Luan, H., Sun, M.: Adversarial training for unsupervised bilingual lexicon induction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1959–1970 (2017)