Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

SPMM: A Soft Piecewise Mapping Model for Bilingual Lexicon Induction

Yan Fan* Chengyu Wang† Boxing Chen‡ Zhongkai Hu§ Xiaofeng He¶

Abstract

Bilingual Lexicon Induction (BLI) aims at inducing word translations in two distinct languages. The generated bilingual dictionaries via BLI are essential for cross-lingual NLP applications. Most existing methods assume that a mapping matrix can be learned to project the embedding of a word in the source language to that of a word in the target language which shares the same meaning. However, a single matrix may not be able to provide sufficiently large parameter space and to tailor to the semantics of words across different domains and topics due to the complicated nature of linguistic regularities. In this paper, we propose a Soft Piecewise Mapping Model (SPMM). It generates word alignments in two languages by learning multiple mapping matrices with orthogonal constraint. Each matrix encodes the embedding translation knowledge over a distribution of latent topics in the embedding spaces. Such learning problem can be formulated as an extended version of the Wahba's problem, with a closed-form solution derived. To address the limited size of training data for low-resourced languages and emerging domains, an iterative boosting method based on SPMM is used to augment training dictionaries. Experiments conducted on both general and domain-specific corpora show that SPMM is effective and outperforms previous methods.

Keywords: bilingual lexicon induction; soft piecewise mapping; Wahba's problem; iterative boosting

1 Introduction

Bilingual Lexicon Induction (BLI) aims at building a translation dictionary between two languages. Such bilingual lexicons are essential for tasks such as cross-lingual information retrieval [9], multilingual POS tagging [28], etc. The most direct application of BLI is machine translation. The generated lexicons are used either as training data for statistical machine translation [25], or as ground truth to incorporate with neural machine translation [4]. For low-resourced languages or emerging domains where parallel corpora are limited, such lexicons are particularly significant because they provide precise word alignments [8].

Most methods exploit bilingual word embeddings to induce bilingual lexicons. Mikolov et al. [22] observe that similar geometric arrangements exist among vector spaces of different languages. They learn a linear matrix to establish the mappings from the source embedding space to the target embedding space. Xing et al. [30] constrain the matrix to be orthogonal with normalized word embeddings. For low-resourced languages or emerging domains, a sufficiently large bilingual lexicon may not be available for training the BLI model. Various approaches are proposed to address this issue by using a small seed dictionary [2], identical words in both languages [26] or with no lexicons at all [17, 31]. Particularly, unsupervised BLI achieves this by formulating the problem as a natural adversarial game [13]. In the game, a generator learns the bilingual mapping matrix. Meanwhile, a discriminator tries to distinguish the languages of words given their embeddings as inputs.

Despite the significant success made in BLI so far, we suggest that the performance can be further improved by addressing the following two problems. i) Word distributions from non-parallel corpora may vary a lot depending on the languages or domains [6]. Learning a single matrix may not be able to capture the translation knowledge across different domains and topics. Consider the example in Fig. 1. We cluster word embeddings trained over Wikipedia corpora. Each word cluster roughly corresponds to a latent topic. Hence, the performance of BLI can be improved by considering multiple, fine-grained mapping matrices. ii) The performance of BLI is highly sensitive to the quality of word embeddings and the amount of training data. As Søgaard et al. [27] point out, the result of BLI is much worse for domain-specific corpora or morphologically rich languages.

In this paper, we propose a BLI model named Soft Piecewise Mapping Model (SPMM). The SPMM

*School of Computer Science and Software Engineering, East China Normal University. E-mail: eileen940531@gmail.com

†School of Computer Science and Software Engineering, East China Normal University. E-mail: chywang2013@gmail.com

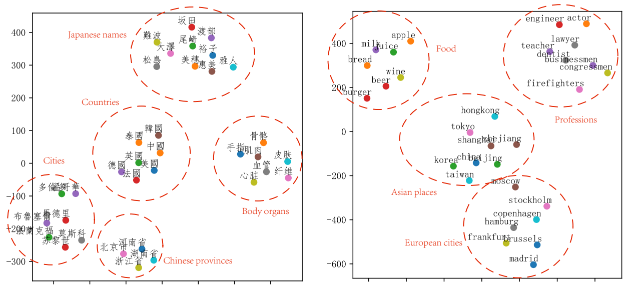
‡Alibaba Group Inc. E-mail: boxing.cbx@alibaba-inc.com

§Alibaba Group Inc. E-mail: zhongkai.hzk@alibaba-inc.com

¶Corresponding author. School of Computer Science and Software Engineering, East China Normal University. E-mail: xfhe@sei.ecnu.edu.cn

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) Language: Chinese

(b) Language: English

Figure 1: The visualization of word embeddings in two languages using t-SNE [19].

method is an extension of Lample et al.'s work [17] by learning multiple mapping matrices with orthogonal constraint. In the initial stage, a clustering algorithm is performed over word embeddings of the source language to discover latent topics. Hence, each word pair in the training set is associated with a weight vector over all latent topics. By injecting probability distributions into the loss function of mapping errors, SPMM solves the mapping problem by minimizing the weighted projection loss with orthogonal constraint over all latent topics. The learning of each matrix is equivalent to an extended version of the *Wahba's problem* [29], to which we derive a closed-form solution based on Singular Value Decomposition (SVD).

Because the size of training dataset (i.e., the training bilingual lexicon) is highly limited in low-resourced languages or emerging domains [6], an iterative boosting method based on SPMM is further introduced. It starts with the initial training of SPMM. In the iterative process, the training lexicon is automatically expanded via a Cross-lingual Self Validation (CSV) strategy. The parameters of SPMM are updated simultaneously.

In summary, we make the following contributions:

- We propose an SPMM method to improve the performance of BLI. An extended *Wahba's problem* is employed to learn multiple mapping matrices, with a closed-form solution derived.
- Based on SPMM, an iterative boosting technique is presented to handle BLI in low-resourced languages or emerging domains.
- We conduct extensive experiments over both general and domain corpora to show that SPMM improves the accuracy of BLI and outperforms previous approaches.

The rest of this paper is as follows. Section 2 summarizes the related work on BLI. We introduce the SPMM approach in detail in Section 3, with experiments presented in Section 4. Finally, we draw the conclusion and discuss the future work in Section 5.

2 Related Work

This section summarizes the related work on BLI. Most studies in this field consist of three steps: i) representing words by low-dimensional embedding vectors; ii) learning mappings between the embedding spaces of different languages; and iii) detecting the correct translation of each word in the source language in the mapped embedding space to obtain a bilingual dictionary.

Early methods for BLI exploit statistical measures to learn the distributional representations of words, such as Pointwise Mutual Information (PMI) [12]. In recent years, deep learning techniques have become the mainstream in the NLP community. Neural language models are proposed to learn low-dimensional representations, such as Word2Vec [23], fastText [5], etc. As discovered by [6, 27], the quality of word embeddings have a great impact on the performance of BLI, especially for long-tail domain words or morphologically rich languages.

The major research focus of BLI is to establish the mappings between the embedding spaces of two different languages. Based on their different problem formulations and learning paradigms, existing methods can be divided into three categories. The first category is to learn cross-lingual embeddings directly. In this category, words in different languages are mapped to a uniform embedding space. Hence, there is no need to learn mapping matrices across languages for BLI. In the literature, several methods modify the Continuous Bag-of-Words (CBOW) model to learn cross-lingual embeddings [7, 11, 14]. The learning objective is to predict center words given the contexts of both languages, with mixed monolingual corpora as inputs.

The second category is to learn mappings to transform words from the source embedding space to the target one. A basic approach is to use a linear mapping matrix [22]. Several constrains are imposed to the matrices for better performance, e.g., the orthogonality of the matrix and the normalization of word embeddings [30, 1, 3]. To lower the requirement of the initial training data, self-learning strategies are proposed to boost the training dictionary for matrix refinement during iterations [2]. For extremely low-resourced languages which do not have initial lexicons to train a mapping matrix, unsupervised BLI approaches have been proposed. Inspired by adversarial game, recent studies learn the mappings without any supervised signals by learning the projection matrices and confusing the discriminator at the same time [31, 17]. Compared to previous methods, our work considers the complicated semantics of natural languages and employs multiple matrices to enhance projection learning in BLI.

The last category is to learn individual mapping matrices for both source and target languages, which

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

project word embeddings of two languages into a third embedding space [16]. It is usually applied to BLI between two low-resourced languages, using English as the "bridging" language.

Once the mapping matrices are learned, the final step is to generate reliable translation pairs. An intuitive way is to find the nearest neighbors of the mapped embeddings in the target space for source words. However, in high-dimensional spaces, this leads to *Hubness problem* [24] where some "hub" vectors are highly likely to be the nearest neighbor of many source words, while others may not be the nearest neighbor of any words. Replacing mean squared loss with max-margin loss function alleviates this problem to some extent [18]. Various methods are proposed to mitigate this problem, such as correcting retrieval scores globally [10], exploiting the earth mover's distance [32], inverting the softmax function [26], and considering cross-domain similarity metrics [17].

3 SPMM: Soft Piecewise Mapping Model

In this section, we introduce the proposed SPMM approach in detail. Preliminaries are provided before we present SPMM in formal. We also introduce how to use the iterative boosting technique based on SPMM.

3.1 Preliminaries In this work, we assume that embeddings of words from source and target languages are trained independently on separate monolingual corpora via fastText [5]. Denote d as the dimension of the embeddings, n as the training dictionary size aligning both languages, and $\vec{x}_i$ as the embedding vector of word x_i.

Mikolov et al. [22] show that the mapping from the source embedding space to the target can be modeled as a linear function by minimizing mapping errors as:

$$\min \sum_{(x_i,y_i)\in D} \|W\vec{x}_i - \vec{y}_i\|^2$$

where $D = \{(x_i, y_i)\}$ is the training set with x_i and y_i being words from the source and target languages that share the same meaning. W is the $d \times d$ mapping matrix that approximates $W\vec{x}_i = \vec{y}_i$.

Xing et al. [30] observe that the performance of BLI can be improved by imposing the orthogonality constraint on W (i.e., $W^T W = I$ where I is the identity matrix). Let X and Y be the $d \times n$ word embedding matrices of source and target languages respectively. This optimization problem, also referred to as the *Procrustes problem* [26], has an exact solution based on SVD:

$$W^* = UV^T, \text{ with } U\Sigma V^T = \text{SVD}(YX^T)$$

3.2 Soft Piecewise Mapping Model In this part, we introduce the problem formulation of SPMM. After that, the parameter learning algorithm is derived, with the dictionary induction technique presented.

3.2.1 Problem Formulation As discussed in the introduction, one mapping matrix tends to be insufficient to model how a word in the source language should be translated to its counterpart in the target language. Hence, SPMM employs multiple matrices to accommodate latent domains or topics in the vocabulary.

A basic approach is to partition the training set based on the latent topics. Each subset of the training set is utilized to learn one single matrix. However, for low-resourced languages or emerging domains, such hard clustering practice is harmful to the BLI performance due to the small size of the training set [6].

In this work, we introduce a *soft piecewise mapping* approach. Let R be a pre-defined number of latent topics, fine-tuned over the validation set. Each pair $(x_i, y_i) \in D$ is associated with a weight vector $< a_{i,1}, a_{i,2}, \cdots, a_{i,R} >$ where $a_{i,r} > 0$ models the degree that the pair (x_i, y_i) belongs to the rth topic. The objective function $J(W_1, W_2, \cdots, W_R)$ is defined as follows:

(3.1)

$$\min J(W_1, W_2, \cdots, W_R) = \frac{1}{2}\sum_{r=1}^{R} \sum_{(x_i,y_i)\in D} a_{i,r}\|W_r\vec{x}_i - \vec{y}_i\|^2$$

$$\text{s.t. } W_r^T \cdot W_r = I, \sum_{i=1}^{|D|} a_{i,r} = 1 \ (r = 1, \ldots, R)$$

where $W_1, \cdots, W_R$ are R mapping matrices. All training data can be used to train every mapping matrix. Each pair (x_i, y_i) is associated with the weight $a_{i,r}$ for learning the rth mapping matrix W_r.

3.2.2 Learning $a_{i,r}$ A direct approach to learn $a_{i,r}$ is through the soft clustering of D, using the distributional representations of (x_i, y_i) as features. We have experimented with several soft clustering algorithms with limited success, e.g., Gaussian Mixture Model (GMM), Fuzzy C-Means (FCM), etc. The possible reasons for the failure are: i) the dimensionality of word embeddings d is high, causing the so-called curse of dimensionality; and ii) the training set D is relatively small. Furthermore, clustering on D only ignores the semantics of words outside the training set.

Here, we present a heuristic clustering method to approximate $a_{i,r}$. Let V_S be the vocabulary set of the source language. We apply K-means to all words in V_S, with their word embeddings as features. Denote $\vec{c}_1, \vec{c}_2, \cdots, \vec{c}_R$ as the R centroid embeddings. $a_{i,r}$ is defined as follows:

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$$a_{i,r} = \frac{\mathrm{sim}(\vec{x}_i, \vec{c}_r) + \gamma}{\sum_{(x_{i'}, y_{i'}) \in D} (\mathrm{sim}(\vec{x}_{i'}, \vec{c}_r) + \gamma)} \tag{3.2}$$

where $\mathrm{sim}(\vec{x}_i, \vec{c}_r)$ is the cosine similarity between the word embeddings $\vec{x}_i$ and the cluster centroid $\vec{c}_r$. Let the adjusting factor $\gamma = 1$ in the experiment to turn the range of the similarity function to positive. Besides this resizing technique, we have also experimented with other alternatives to adjust the cosine value such as the exponential function $\exp(\mathrm{sim}(\vec{x}_i, \vec{c}_r))$ and sigmoid function $\sigma(\mathrm{sim}(\vec{x}_i, \vec{c}_r))$.

3.2.3 Learning W_r After all the values of $a_{i,r}$ are fixed, we aim at learning all the mapping matrices W_r. Denote the optimal solution to Eq. (3.1) as $\{W_1^*, W_2^*, \cdots, W_R^*\}$.

LEMMA 3.1. *The value of W_r^* is the optimal solution to Eq. (3.3):*

$$\begin{aligned} &\min J(W_r) = \frac{1}{2} \sum_{(x_i, y_i) \in D} a_{i,r} \| W_r \vec{x}_i - \vec{y}_i \|^2 \\ &s.t.\ W_r^T \cdot W_r = I \end{aligned} \tag{3.3}$$

It is trivial to prove Lemma 3.1 because the optimization process of each matrix W_r is independent from each other. Hence, we omit the details here.

The optimization of Eq. (3.3) is equivalent to an extended version of the *Wahba's problem* [21] in applied mathematics. It is extensively applied to process three-dimensional vector observations between two coordinate systems for satellite attitude determination. In this work, we extend an SVD-based solution to the original *Wahba's problem* [20] to d dimensions ($d > 3$).

THEOREM 3.1. *The d-dimensional* Wahba's problem *can be solved by the following SVD-based solution:*

1. $B_r = \sum_{(x_i, y_i) \in D} a_{i,r} \vec{y}_i \cdot \vec{x}_i^T$;
2. $SVD(B_r) = U_r \Sigma_r V_r^T$;
3. $R_r = diag(\underbrace{1, \ldots, 1}_{d-1}, det(U_r) det(V_r))$;
4. $W_r^* = U_r R_r V_r^T$.

We extend Markley's work [20] to prove the correctness of Theorm 3.1 as follows:

Proof. Because Eq. (3.2) ensures $\sum_{i=1}^{|D|} a_{i,r} = 1$, based on [29], we re-write Eq. (3.3) as follows:

$$J_r = 1 - \sum_{(x_i, y_i) \in D} a_{i,r} \cdot \vec{y}_i^T W_r \vec{x}_i = 1 - \mathrm{tr}(W_r B_r^T) \tag{3.4}$$

where $B_r = \sum_{(x_i, y_i) \in D} a_{i,r} \vec{y}_i \cdot \vec{x}_i^T$. According to Theorem 3.1, we decompose B_r based on SVD: $\mathrm{SVD}(B_r) = U_r \Sigma_r V_r^T$. $\Sigma_r = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$.

Based on the result of SVD, we construct two $d \times d$ orthogonal matrices and a $d \times d$ diagonal matrix:

$$U_r^+ = U_r \mathrm{diag}(\underbrace{1, \ldots, 1}_{d-1}, \det(U_r))$$

$$V_r^+ = V_r \mathrm{diag}(\underbrace{1, \ldots, 1}_{d-1}, \det(V_r))$$

$$\Sigma_r^{'} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{d-1}, \lambda_d \det(U_r) \det(V_r))$$

Because U_r and V_r are orthogonal matrices, $\det(U_r)\det(V_r) = \pm 1$. B_r is re-decomposed by U_r^+, V_r^+ and $\Sigma_r^{'}$, i.e., $B_r = U_r^+ \Sigma_r^{'} {V_r^+}^T$. Let $M_r = {U_r^+}^T W_r V_r^+$ be a $d \times d$ auxiliary matrix. Based on the cyclic invariance property of the trace, we have:

$$\begin{aligned} \mathrm{tr}(W_r B_r^T) &= \mathrm{tr}(W_r V_r^+ \Sigma_r^{'} {U^+}^T_r) \\ &= \mathrm{tr}(\Sigma_r^{'} {U^+}^T_r W_r V_r^+) = \mathrm{tr}(\Sigma_r^{'} M_r) \end{aligned} \tag{3.5}$$

Substituting Eq. (3.5) into Eq. (3.4), we obtain: $J_r = 1 - \mathrm{tr}(\Sigma_r^{'} M_r)$. According to the Euler angle parameterization, we can see that J_r is minimized when $M_r = I$. Hence, the optimal solution to Eq. (3.3) is derived as:

$$W_r^* = U_r^+ {V_r^+}^T = U_r \mathrm{diag}(\underbrace{1, \ldots, 1}_{d-1}, \det(U_r)\det(V_r)) {V_r}^T \blacksquare$$

3.2.4 Training Algorithm Algorithm 1 summarizes the high-level training procedure of SPMM. The inputs to the algorithm are a collection of the source language vocabulary V_S, a bilingual training set $D = \{(x_i, y_i)\}$, parameter R and two pre-trained neural language models over both source and target languages. The outputs of SPMM include cluster centroids $\vec{c}_1, \vec{c}_2, \cdots, \vec{c}_R$ and mapping matrices $W_1^*, W_2^*, \cdots, W_R^*$.

Algorithm 1 Training Algorithm of SPMM

```
($\vec{c}_1, \ldots, \vec{c}_R$)=K-means($V_S$);
for each $(x_i, y_i) \in D$ do
    for $r = 1$ to $R$ do
        Compute $a_{i,r}$ based on Eq. (3.2);
    end for
end for
for $r = 1$ to $R$ do
    Minimize Eq. (3.3) according to Theorem 3.1, with
    the optimal solution as $W_r^*$;
end for
return $\vec{c}_1, \vec{c}_2, \cdots, \vec{c}_R$ and $W_1^*, W_2^*, \cdots, W_R^*$.
```

3.2.5 Dictionary Induction Once the model is trained, it can find the most probable translation of the target language for a new word of the source language.

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Given a source word x_i, SPMM computes the weights $a_{i,r}$ based on Eq. (3.2), and projects it to the embedding space of the target language:

$$\tilde{y}_i = \sum_{r=1}^{R} a_{i,r} W_r \vec{x}_i \tag{3.6}$$

The embedding vector of the most probable translation of x_i should be closest to the mapped embedding $\tilde{y}_i$, in terms of cosine similarity.

However, the "nearest neighbor" technique often leads to the *Hubness problem* in the high-dimensional space [24, 15]. It refers to the situation where a "hub" embedding vector is the nearest neighbor of many others while it can only refer to the correct translation of one source word. This problem has also been pointed out and addressed in other works, such as mapping learning based on the max-margin ranking loss [18], the inverted softmax technique [26].

In this paper, we apply Cross-domain Similarity Local Scaling (CSLS) [17] to the SPMM approach. Due to the asymmetry of vector spaces, if $\vec{x}_i$ is the nearest neighbor of $\vec{y}_i$, $\vec{y}_i$ is not necessarily the nearest neighbor of $\vec{x}_i$. The idea of CSLS is to make isolated word vectors more possible to be the nearest neighbors and to lower the probability of word vectors in the "dense" region being selected as nearest neighbors.

Denote $N_T(\tilde{y}_i)$ as the collection of top-m nearest neighbors of a mapped embedding vector $\tilde{y}_i$ in the target language. Let $N_S(\vec{y}_i)$ be the collection of top-m nearest neighbors of an embedding vector $\vec{y}_i$ in the source language. The similarity measure CSLS between the mapped embedding vector $\tilde{y}_i$ and the embedding of a word of the target language $\vec{y}_i$ can be computed as follows:

$$\begin{aligned} & CSLS(\tilde{y}_i, \vec{y}_i) = 2 \cdot \mathrm{sim}(\tilde{y}_i, \vec{y}_i) \\ & - \frac{1}{m} \sum_{\vec{y} \in N_T(\tilde{y}_i)} \mathrm{sim}(\tilde{y}_i, \vec{y}) - \frac{1}{m} \sum_{\vec{y} \in N_S(\vec{y}_i)} \mathrm{sim}(\vec{y}_i, \vec{y}) \end{aligned} \tag{3.7}$$

where $\mathrm{sim}(\tilde{y}_i, \vec{y}_i)$ is the cosine similarity between the word embeddings. The last two subtracted items averages the similarities of m nearest neighbors of the embedding spaces for both source and target languages.

In summary, during the dictionary induction stage, given a previously unseen word x_i of the source language, SPMM predicts the mapped embedding vector $\tilde{y}_i$ based on Eq. (3.6). Next, it generates the top-m most probable translations of the target language by m-nearest search using CSLS in Eq. (3.7).

3.3 Iterative Boosting Technique

This technique is an iterative process that expands the size of the training set D without human supervision. It is typically useful for BLI over low-resourced languages and emerging domains with a small training set.

For each iteration, we expand the training dictionary as follows. We first sample two collections of words (denoted as D_S and D_T) from the source and target languages, respectively. Each word x_i in D_S is randomly drawn from the entire vocabulary set with probability $\propto$ *count*(x_i) where *count*(x_i) is the frequency count in large text corpora. D_T is constructed in the same way. Heuristically, we constrain that $|D_S| = |D_T| = m$. We consider frequent words here due to the high qualities of their embeddings [17].

To extract word pairs (x_i, y_i) that are likely to share the same meaning without human supervision, we present a Cross-lingual Self Validation (CSV) strategy. Denote $P_T(x_i)$ as the collection of top-m most probable translations of the target language w.r.t. the word $x_i \in D_S$, predicted by SPMM. Symmetrically, $P_S(y_i)$ is the collection of top-m most probable translations of the source language w.r.t. the word $y_i \in D_T$. We treat word pairs in the following collection D^* as high-confidence predictions to be added to the training set:

$$\begin{aligned} D^* = & \{(x_i, y_i) | x_i \in D_S, y_i \in P_T(x_i)\} \\ & \cap \{(x_i, y_i) | y_i \in D_T, x_i \in P_S(y_i)\} \end{aligned} \tag{3.8}$$

The reasons that the translations in D^* are likely to be correct are twofolds: i) both words in the pair $(x_i, y_i) \in D^*$ are likely to be frequent with high-quality embeddings; and ii) the correctness of the pair is self-validated from both language translation directions.

After the collection D^* is generated, we merge D and D^* to train SPMM again. This process iterates until the precision stops to increase over the validation set. Finally, we summarize the iterative boosting technique in Algorithm 2.

Algorithm 2 Iterative Boosting Technique

```
Initialize $D^* = \emptyset$;
repeat
    Train SPMM over $D \cup D^*$ by Algorithm 1;
    Sample word collections $D_S$ and $D_T$;
    Generate word collection $D^*$ based on Eq. (3.8);
    Update $D = D \cup D^*$;
until Precision stops to increase over validation set
```

4 Experiments

In this section, we conduct extensive experiments to evaluate the performance of SPMM. We also compare it with state-of-the-art to make the convincing conclusion.

4.1 Experiments over General Corpora

This set of experiments evaluate the performance of SPMM over general-domain corpora for BLI.

4.1.1 Datasets and Experimental Settings

For fair comparison, we use fastText word embeddings of

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Table 1: Performance of bilingual word translation in terms of Precision@K. (Language abbreviations: en: English, es: Spanish, fr: French, de: German, it: Italian, zh: Chinese, vi: Vietnamese) (%)

Language Pair	en-es			es-en			en-fr			fr-en		
Precision@K	1	5	10	1	5	10	1	5	10	1	5	10
Mikolov et al. [22]	72.3	85.6	89.0	74.1	87.8	90.9	68.6	85.3	88.7	71.3	85.9	88.9
Dinu and Baroni [10]	73.3	87.7	90.7	73.2	87.9	92.0	72.0	87.1	90.3	72.7	86.9	90.7
Artetxe et al. [2]	78.1	88.7	90.6	81.6	89.9	92.7	78.2	88.9	91.3	76.3	87.7	91.6
Smith et al. [26]	80.1	89.5	91.9	80.3	91.1	93.4	77.8	89.0	92.1	78.2	89.2	91.8
Lample et al. [17]	81.4	91.3	93.5	82.9	91.9	**94.1**	81.1	90.8	**92.9**	82.4	**91.7**	93.3
SPMM (No-iter)	81.3	91.2	93.5	82.6	**92.1**	94.0	80.7	90.8	**92.9**	82.5	91.5	93.4
SPMM (Full)	**81.9**	**91.7**	**93.7**	**83.7**	**92.1**	94.0	**81.4**	**91.2**	**92.9**	**82.7**	**91.7**	**93.7**
Language Pair	en-de			de-en			en-it			it-en		
Precision@K	1	5	10	1	5	10	1	5	10	1	5	10
Mikolov et al. [22]	61.1	81.9	86.3	61.7	77.9	82.5	65.1	81.7	85.6	68.6	83.3	86.6
Dinu and Baroni [10]	62.9	84.3	88.7	63.6	79.9	84.7	67.5	84.9	88.2	68.4	84.1	88.1
Artetxe et al. [2]	70.2	86.4	89.8	69.5	82.3	86.5	72.7	85.3	89.1	74.1	85.6	88.3
Smith et al. [26]	72.0	87.9	90.9	70.8	84.5	88.1	74.3	87.4	90.7	75.9	87.1	89.9
Lample et al. [17]	73.5	89.3	92.0	**72.4**	**86.1**	88.3	76.2	88.8	91.6	77.9	88.2	**90.7**
SPMM (No-iter)	73.1	89.6	92.1	**72.4**	86.0	**88.5**	75.8	88.8	91.7	77.3	88.2	90.6
SPMM (Full)	**74.1**	**89.6**	**92.2**	**72.4**	86.0	**88.5**	**76.4**	**88.9**	**92.0**	**78.3**	**88.4**	90.6
Language Pair	en-zh			zh-en			en-vi			vi-en		
Precision@K	1	5	10	1	5	10	1	5	10	1	5	10
Mikolov et al. [22]	14.0	25.0	29.0	30.1	53.3	60.6	9.5	23.4	29.9	25.7	45.4	53.6
Dinu and Baroni [10]	20.5	41.5	51.1	31.0	53.5	60.6	23.3	50.5	60.7	43.7	68.7	76.5
Artetxe et al. [2]	9.1	20.4	26.9	15.1	27.7	34.3	13.5	27.2	34.3	34.2	57.2	76.3
Smith et al. [26]	40.1	57.3	62.9	33.5	55.2	63.6	33.1	50.9	59.3	47.7	69.7	76.9
Lample et al. [17]	32.4	55.0	62.5	36.7	58.4	65.3	41.3	58.6	64.0	55.3	73.5	79.4
SPMM (No-iter)	**42.6**	**60.1**	64.4	36.6	58.6	**65.4**	41.5	58.7	64.1	55.3	73.4	79.4
SPMM (Full)	**42.6**	**60.1**	**64.5**	**38.0**	**59.0**	**65.4**	**43.0**	**59.7**	**64.5**	**59.1**	**76.2**	**80.8**

six language pairs released in [17]. The embeddings are trained over Wikipedia corpora with 300 dimensions.

To evaluate the effectiveness of SPMM, we conduct the experiments over BLI. It aims at finding the correct translations in target language given a set of source words. We utilize Precision@K ($K = 1, 5, 10$) as the evaluation metrics, to compute the precision of top-K retrieved candidate words. The ground truth bilingual dictionaries that we use are publicly available in [17]. For each language pair, we use 5,000 unique source words and their translations and 1,500 for testing.

4.1.2 General Performance The results of the bilingual word translation task for six language pairs in terms of Precision@K are summarized in Table 1. We consider the following baselines:

- Mikolov et al. [22]: It learns a single mapping matrix without any constraints.
- Dinu and Baroni [10]: It addresses the *Hubness problem* by using a globally-corrected approach for dictionary induction, instead of the nearest neighbor retrieval technique in [22].
- Artetxe et al. [2]: It improves the work [22] by using a mapping matrix with orthogonal constraint and an iterative technique.
- Smith et al. [26]: It uses the inverted softmax function for the retrieval of target words.
- Lample et al. [17]: It improves the work [2] by proposing the cross-domain similarity local scaling (CSLS) method to address the *Hubness problem*.
- SPMM (No-iter): It is the variant of SPMM without the iterative boosting technique.

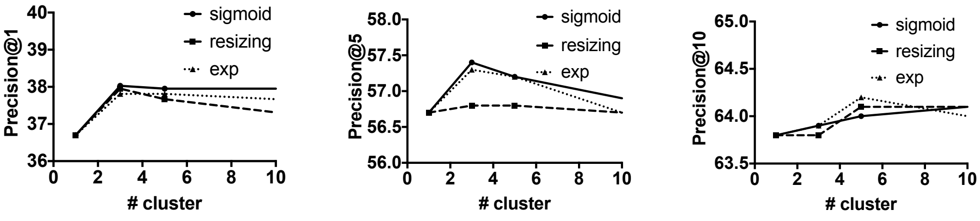
In the experiments, we obtain the original codes of all the baselines from other papers and produce the results by ourselves. For our method, we have the following default parameter settings: $R = 3$, $m = 10000$ and run our algorithm in 5 iterations. We also tune the parameters of SPMM for further parameter analysis.

From the results in Table 1, we can see that SPMM generally outperforms all the baselines on all six language pairs. The results of the first two baselines [22, 10] can not compete with the rest of approaches. They simply use gradient descent to learn mapping matrices by minimizing mapping errors. It shows that adding orthogonal constraint on mapping matrices improves the performance of BLI persistently. As for the strategy to address the *Hubness problem*, the CSLS technique used in both Lample et al. [17] and SPMM is the most effective, compared to the globally-corrected approach [10] and the inverted softmax method [26]. The iterative training algorithm in Smith et al. [26] can improve the performance, but not as much as SPMM (Full). Although the iterative version does not always boost the performance of SPMM (No-iter), our dictionary boost-

Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) Precision@1 (%) (b) Precision@5 (%) (c) Precision@10 (%)

Figure 2: Parameter analysis of the non-iterative version of SPMM.

ing strategy gains significant improvements for language pairs such as Vietnamese-English. As for SPMM, it generally outperforms other baseline models, especially for the language pair of English-Chinese, gaining over 10 percents in terms of Precision@1. Therefore, we can see that for language pairs that belong to different language families (e.g., English and Chinese), word distributions in corpora are more likely to be different. It is necessary to map the embeddings with multiple matrices in a fine-grained manner.

4.1.3 Parameter Analysis To show how different values of parameters can effect the performance of SPMM, we conduct the following experiments over the Chinese-English language pair. In Fig. 2, we present the word translation performance of the non-iterative version of the SPMM method. We tune the cluster number R from 1 to 10. When $R = 1$, SPMM is down-graded to a single matrix learning method. It can be seen that SPMM achieves the highest performance when $R = 3$ in terms of Precision@1,5. We also vary the similarity function Eq. (3.2) as sigmoid, resizing and exponential functions to test their effectiveness. It shows that the sigmoid similarity function is generally the most suitable for SPMM when $R = 3$.

We further study whether the CSV technique used in the iterative boosting process can improve the translation performance. We compare two recent BLI methods which include an iterative learning process: Lample et al. [17] and SPMM. We run both models in five iterations, and report the performance in each iteration in Fig. 3. The performance of the iterative version of Lample et al. [17] drops significantly because it uses all generated word pairs as the training set in the next iteration. Errors caused by model prediction are likely to propagate. In contrast, SPMM keeps ground-truth datasets in the training set in all iterations and employs the CSV technique to guarantee the high quality of the generated word pairs. In Table 1, the Precision@1, Precision@5 and Precision@10 scores are improved by 1.3, 0.6 and 0.1 percents in the first five iterations.

4.2 Experiments over the Medical Domain

To test whether the SPMM method can deal with BLI in specific domains, we conduct experiments for rare words translation over the medical domain.

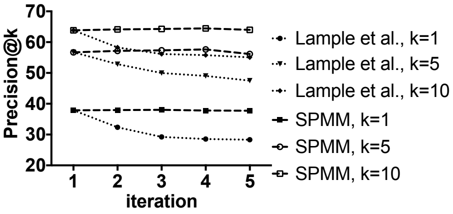


Figure 3: Performance comparison between Lample et al. [17] and SPMM in five iterations. (%)

4.2.1 Dataset and Experimental Settings The medical bilingual lexicon that we use is taken from [6]. A medical text corpus is created by crawling the titles of medical Wikipedia articles, medical term-pairs, patents, documents from the European Medicines Agency, consisting 3,108,183 sentences in English and German. A medical rare word bilingual lexicon is created by randomly sampling words occurring between 3 and 5 times in the corpus. In total, there are 8,079 medical rare word pairs in English and German. For medical BLI, we aim at translating English medical terms into German. We spilt the dataset into 6,079 training, 1,000 validation and 1,000 testing.

4.2.2 Result Analysis In the experiments, we use the same settings and baselines as in the experiments of general copora, and summarize the results in Table 2. The results show that SPMM gains more significant improvements in the medical domain data, compared to the results of general corpora. As seen also in Table 2, the Precision@1, Precision@5 and Precision@10 scores are improved by 1.2, 2.1 and 2.3 percents. In contrast, for the same English-German language pair in general domain, the Precision@1, Precision@5 and Precision@10 scores are improved by 0.6, 0.3 and 0.2 percents only. Therefore, multiple mapping matrices employed by SPMM are proved to be effective for word translation task especially for domain words with low-frequencies. In this experiment, the iterative learning strategy, SPMM (Full) does not obtain significant improvements with 0.3 percent of improvements in terms of Precision@1. This is because the size of the training

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Table 2: Performance of bilingual medical word translation in terms of Precision@K. (%)

Medical	P@1	P@5	P@10
Mikolov et al. [22]	17.1	33.6	39.0
Dinu and Baroni [10]	21.0	38.9	46.1
Artetxe et atl. [2]	28.9	41.3	45.3
Smith et al. [26]	30.7	44.0	49.6
Lample et al. [17]	36.4	46.8	50.9
SPMM (No-iter)	37.3	**48.9**	**53.2**
SPMM (Full)	**37.6**	**48.9**	**53.2**

dictionary used in this task is relatively large. It does not give much space for improvements even if we use the iterative boosting strategy.

4.3 Industrial Experience in E-commerce

In this section, we report our work on the e-commerce domain in Alibaba Group Inc. In Alibaba, there exist millions of fine-grained product names that need to be translated to other languages to support internationalization. However, the accurate translation of such product names is challenging because they contain a large amount of fine-grained, domain specific expressions. It is difficult for humans to annotate a large training set.

In this work, we aim at translating product names from Chinese to English. To evaluate the performance, we ask human annotators to create a bilingual lexicon, spilt into 530 training, 250 validation and 245 testing. Word embeddings in Chinese is trained on 4000K sentences of product description from Tmall.com, containing 70K unique words. For English word embeddings, we train the fastText model on 60M sentences of product description from Alibaba.com, and finally obtain 680K unique words. Since the work of Lample et al. [17] is the state-of-the-art, we only compare the performance of iterative SPMM and Lample et al. [17].

Table 3 illustrates the performance of both models in 30 iterations. It shows SPMM outperforms Lample et al. [17]. As for the iterative version of Lample et al. [17], we see the same trend as in Fig. 3, where the performance does not improve with more iterations. In this experiment, the precision is not strictly decreasing similar to Fig. 3, but with some fluctuations. The performance of iterative SPMM, however, has a large improvement with more iterations. For the e-commerce corpus, the Precision@1, Precision@5 and Precision@10 scores are improved by 3.7, 4.5 and 9.4 percents. This improvement is much higher than the result of medical domain, which further illustrates that the iterative boosting strategy is very effective for the situation where the training dictionary is small in size.

Table 3: Performance of e-commerce word translation in terms of Precision@K. (%)

E-commerce	Iteration	P@1	P@5	P@10
Lample et al. [17]	1	12.2	25.7	30.2
	5	9.8	21.2	31.0
	10	10.2	24.5	29.4
	20	10.6	23.2	30.6
	30	9.7	22.9	31.4
SPMM	1	13.9	29.4	35.1
	5	15.1	29.0	36.3
	10	15.5	29.0	38.8
	20	**15.9**	26.9	38.4
	30	14.7	**30.2**	**39.6**

Table 4: Case studies of e-commerce word translation. For each Chinese e-commerce term, top-5 results generated by SPMM and Lample et al. [17] are listed, with the correct translations printed in bold.

Term	Lample et al. [17]	SPMM
接触器 (contactor)	1.2kv	relay
	tolerance	**contactor**
	1μf	cdc1
	89a	telemecanique
	watts	nais
摄影头 (lens)	receiver	webcam
	tv	camera
	analogtv	**lens**
	adio	megapixels
	dvbt	2.0mp
指示器 (indicator)	unplasticized	signboard
	number	sign
	nbc-288	**indicator**
	tagboard	traffic
	indicator	signpost

4.3.1 Case Studies

We specify top-5 English candidate words for three examples of Chinese product names in Table 4. As seen, correct translations rank higher in SPMM. For “接触器 (contactor)”, SPMM finds manufacturing companies such as Telemecanique and NAIS, while Lample et al. [17] rank property names (watts), property values (1.2kv) and product types (89a) higher. In the case of “摄影头 (lens)”, we also notice that SPMM finds words related to camera correctly, while Lample et al. [17] target at devices related to TV. The results is even obvious for word “指示器 (indicator)”, where top-5 translations of SPMM is much more relevant compared to results of Lample et al. [17].

5 Conclusion and Future Work

This paper introduces a Soft Piecewise Mapping Model (SPMM) to improve the performance of BLI. It learns word alignments in two languages by multiple mapping matrices with orthogonal constraint. We further address the BLI task for low-resourced languages and emerging domains by an iterative boosting technique. Experiments illustrate the effectiveness of the proposed approach. In the future, we aim at improving our work for unsupervised neural machine translation.

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 05/08/19 to 104.199.184.177. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Acknowledgements

This work is supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904. It was partly done when Yan Fan visited the Alibaba Group. Yan Fan would also like to thank Weihua Luo, Yangbin Shi, Jiayi Wang, Jun Lu and Xiaoyu Lv for the support of her research.

References

[1] M. Artetxe, G. Labaka, and E. Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *EMNLP*, pages 2289–2294, 2016.

[2] M. Artetxe, G. Labaka, and E. Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *ACL*, pages 451–462, 2017.

[3] M. Artetxe, G. Labaka, and E. Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *AAAI*, 2018.

[4] P. Arthur, G. Neubig, and S. Nakamura. Incorporating discrete translation lexicons into neural machine translation. In *EMNLP*, pages 1557–1567, 2016.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017.

[6] F. Braune, V. Hangya, T. Eder, and A. M. Fraser. Evaluating bilingual word embeddings on the long tail. In *NAACL*, pages 188–193, 2018.

[7] H. Cao, T. Zhao, S. Zhang, and Y. Meng. A distribution-based model to learn bilingual word embeddings. In *COLING*, pages 1818–1827, 2016.

[8] T. Cohn, S. Bird, G. Neubig, O. Adams, and A. J. Makarucha. Cross-lingual word embeddings for low-resource language modeling. In *EACL*, 2017.

[9] J. Dadashkarimi, A. Shakery, and H. Faili. A probabilistic translation method for dictionary-based cross-lingual information retrieval in agglutinative languages. *CoRR*, abs/1411.1006, 2014.

[10] G. Dinu and M. Baroni. Improving zero-shot learning by mitigating the hubness problem. *CoRR*, abs/1412.6568, 2014.

[11] L. Duong, H. Kanayama, T. Ma, S. Bird, and T. Cohn. Learning crosslingual word embeddings without bilingual corpora. In *EMNLP*, pages 1285–1295, 2016.

[12] É. Gaussier, J. Renders, I. Matveeva, C. Goutte, and H. Déjean. A geometric view on bilingual lexicon extraction from comparable corpora. In *ACL*, pages 526–533, 2004.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, volume 2672-2680, 2014.

[14] S. Gouws, Y. Bengio, and G. Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. In *ICML*, pages 748–756, 2015.

[15] H. Jegou, C. Schmid, H. Harzallah, and J. J. Verbeek. Accurate image search using the contextual dissimilarity measure. *TPAMI*, 32(1):2–11, 2010.

[16] H. Kanayama, T. Cohn, T. Ma, S. Bird, and L. Duong. Multilingual training of crosslingual word embeddings. In *EACL*, pages 894–904, 2017.

[17] G. Lample, A. Conneau, L. Denoyer, H. Jégou, et al. Word translation without parallel data. In *ICLR*, 2018.

[18] A. Lazaridou, G. Dinu, and M. Baroni. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *ACL-IJCNLP*, pages 270–280, 2015.

[19] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[20] F. L. Markley. Attitude determination using vector observations and the singular value decomposition. *Journal of the Astronautical Sciences*, 36(3):245–258, 1988.

[21] F. L. Markley and J. L. Crassidis. *Fundamentals of spacecraft attitude determination and control*, volume 33. Springer, 2014.

[22] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[24] M. Radovanovic, A. Nanopoulos, and M. Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *JMLR*, 11:2487–2531, 2010.

[25] S. H. Ramesh and K. P. Sankaranarayanan. Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In *NAACL*, pages 112–119, 2018.

[26] S. L. Smith, D. H. P. Turban, S. Hamblin, and N. Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *ICLR*, 2017.

[27] A. Søgaard, I. Vulic, and S. Ruder. On the limitations of unsupervised bilingual dictionary induction. In *ACL*, pages 778–788, 2018.

[28] O. Täckström, D. Das, S. Petrov, R. T. McDonald, and J. Nivre. Token and type constraints for cross-lingual part-of-speech tagging. *TACL*, 1:1–12, 2013.

[29] G. Wahba. A least squares estimate of satellite attitude. *SIAM review*, 7(3):409–409, 1965.

[30] C. Xing, D. Wang, C. Liu, and Y. Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *NAACL*, 2015.

[31] M. Zhang, Y. Liu, H. Luan, and M. Sun. Adversarial training for unsupervised bilingual lexicon induction. In *ACL*, pages 1959–1970, 2017.

[32] M. Zhang, Y. Liu, H. Luan, M. Sun, T. Izuha, and J. Hao. Building earth mover's distance on bilingual word embeddings for machine translation. In *AAAI*, pages 2870–2876, 2016.

Copyright © 2019 by SIAM
Unauthorized reproduction of this article is prohibited