Inverted Bilingual Topic Models for Lexicon Extraction from Non-parallel Data

Tengfei Ma IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
Tengfei.Ma1@ibm.com

Abstract

A good lexicon is an important resource for various cross-lingual tasks such as information retrieval and text mining. In this paper, we focus on extracting translation pairs from non-parallel cross-lingual corpora. Previous lexicon extraction algorithms for non-parallel data generally rely on an accurate seed dictionary and extract translation pairs by using context similarity. However, there are two problems. One, a lot of semantic information is lost if we just use seed dictionary words to construct context vectors and obtain the context similarity. Two, in practice, we may not have a clean seed dictionary. For example, if we use a generic dictionary as a seed dictionary in a special domain, it might be very noisy. To solve these two problems, we propose two new bilingual topic models to better capture the semantic information of each word while discriminating the multiple translations in a noisy seed dictionary. We then use an effective measure to evaluate the similarity of words in different languages and select the optimal translation pairs. Results of experiments using real Japanese-English data demonstrate the effectiveness of our models.

1 Introduction

The rapid growth of the internet, massive amounts of multilingual information have been available on the different information channels. The number of non-English pages is rapidly expanding. According to the report of March 2015 there are 49.4% of the websites written in non-English languages; and this number is still increasing because the growth rate of English websites is much lower than many other languages such as Spanish, Chinese or Arabic. In this multi-language environment, one challenging but desirable task is to integrate the information in different languages.

Bilingual lexicons play an important role in cross-lingual information retrieval and text mining tasks. However, there are often no existing dictionary for some technical data or low-resourced language pairs. Creating a good bilingual lexicon costs a lot, so automatic lexicon extraction has long been studied.
in the area of natural language processing. For example, extracting translation pairs in a special domain \cite{1,3} has attracted a lot of attention. There are always novel words or new expressions emerging, and a generic dictionary can hardly keep up with these. Therefore, various methods for lexicon extraction have been proposed.

Previous approaches to lexicon extraction have depended either on a parallel/comparable data corpus or a seed dictionary. In practice, we often do not have parallel/comparable data, so the most common way is to use the context similarity to extract lexicons. It is assumed that translation pairs tend to occur in similar contexts across languages, which means we can first extract the context for each word and assign them to a degree of association. By regarding the words in seed dictionaries as pivot words, we can then calculate the similarity between different context vectors across languages.

The context vectors are high-dimensional and very sparse. Words in different languages are connected via only pivot words, while other words in the context vectors are ignored. A simple similarity measure between these vectors, such as cosine similarity, might not capture enough semantic relatedness. Moreover, these methods largely depend on the quality of the seed dictionary. However, when we use a generic dictionary as the seed dictionary for a special domain, it is very noisy because some of the translations may not apply to this domain. A word may have several translations, but it is possible that only one of them is the correct meaning in a target corpus. For example, the Japanese word "立会い" has four translations in a generic dictionary: [presence, attendance, session, market]; but in the law domain, the last two translations are highly depreciated. The translation word "market" is in fact more likely to be connected with "市場" than "立会い". If we include the pair "(立会い, market)", it could be regarded as noise.

In this paper, we propose the utilization of topic models to better measure the semantic relatedness and resolve the noise problem in a seed dictionary. Bilingual topic models have been successfully used for lexicon extraction from comparable data \cite{20}. However, it has never been applied to non-aligned data because in a topic model, we can only represent the topic distributions for documents, and it is hard to integrate the word relationship into the model. In contrast, document relationship is easily modeled by this kind of model \cite{6,5}.

Considering this feature of topic models, we develop a new approach to topic modeling by reversing the roles of documents and words in a topic model. We first represent each word as a pseudo document and then model the words instead of the original documents. In this work, we use inverted indexing to represent a word as a list of documents in which it occurs. After obtaining the pseudo documents, we use our topic models to model each word as a topic distribution. Different from the motivation of previous work related to cross-lingual inverted indexing \cite{18}, we do not consider connections between documents but only between words. Each translation pair is assumed to own the same topic distribution. In this way, the topics in different languages can also be connected.

Next, in order to solve the problem of noisy translations in the seed dictionary, we add a new hierarchy in our models to integrate the probability of
translations. The translations in the seed dictionary are not always regarded as true. Instead, they are selected with a probability based on the topic similarities. In addition, our model is semi-supervised, as we only have a subset of words translated, and the remaining words do not have any connection with words in other languages. This means we can utilize all the data instead of only the connected data that is modeled in the original Bilingual LDA. We use Gibbs sampling for posterior inference. Once we derive the topic distributions for each word, we can get the similarity between words across languages on the basis of their topic distributions. In contrast to conventional cosine similarity and KL divergence, we define our similarity measure as the probability of a word generating another. Given a word in a source language, the word with the most similar topic distributions in the target language is then regarded as its translation.

The main contributions of this paper are as follows:

• We advance a new framework of lexicon extraction by combing the idea of inverted indexing and topic models.

• The new framework uses new topic models that extends the classical Bilingual LDA from two major aspects:
  – 1) incorporating all words into the model instead of using only connected words
  – 2) allowing multiple translations and modeling the probability of each.

• We define a similarity measure of two words across languages from the conditional generating probability and demonstrate through experiments that it outperforms other measures.

The following sections are organized as follows: we first review related work in Section 2. Then we propose our problem definition and new models in Section 3. Section 4 explains how we use our topic model to measure the word similarity. In Section 5 we describe our experiments. At last we conclude our work in Section 6.

2 Related Work

The most well-established work on lexicon extraction is based on word alignment in parallel datasets. We can easily use a statistical machine translation system [11] to induce translation pairs from parallel data. However, parallel data is not plentiful for all language pairs or all domains. This restricts the usefulness of these methods.

Some researchers have relaxed the requirement of parallel data, utilizing document-comparable data such as that found on Wikipedia. Vulić et al. [20] developed a bilingual topic model to obtain cross-lingual topics, and used them to compute the similarity between words in different languages. Recently, word
embedding [21] has also been extended to the cross-lingual scenario by using comparable data to derive the translations. Document pairs are merged to form a pseudo new document, and thus words in different languages finally lie on the same space.

In contrast to these types, our work focuses on extracting special dictionaries from non-parallel data. Lexicon extraction from non-parallel data was pioneered by [16] and [7]. Instead of parallel/comparable documents, they use a seed dictionary as the pivots. Generally, this kind of approach can be factorized into two steps: 1, construct a context vector for each word, and 2, compute the context similarities on the basis of pivot words (i.e., seed dictionary entries). A common hypothesis is that a word and its translation tend to occur in similar contexts. Previous research has defined various correlation measures to construct a context vector representation for a word, including tf-idf [7] and pointwise mutual information (PMI) [1]. As for the similarity computation, cosine similarity [7], non-aligned signatures (NAS) [17], and Johnson-Shannon divergence [15], etc. can be used.

The context similarity-based models rely on the quality and the size of seed dictionaries. When a seed dictionary is small, the context vector will be too sparse and the similarity measure is not accurate enough. Recent work has used graph-based methods to propagate the seed dictionaries [10, 19]. There are also some methods that project the word vectors in different languages into the same low-dimensional space, such as linear transformation for cross-lingual word embedding [12]. Our motivation is similar to the graph-based and word embedding-based models in that we use a topic model to represent each word as a topic distribution in order to avoid the sparseness of context vectors. However, while the previous approaches generally just select the reliable translations as seeds [12, 9], we assume our seed dictionary is noisy. We add the probability of existing translations as a new latent variable and make our model more robust and generalizable.

3 Proposed Model

In this section, we start by making a formal definition of our problem. Then we will describe the details of our new model, following a brief introduction to the background knowledge of cross-lingual topic models.

3.1 Problem Definition

Assume that we are given only two mono-lingual corpora in different languages, $C^e$ and $C^j$. They are neither sentence-aligned nor document-aligned, but are in the same domain. The documents in $C^e$ are noted as $\{d^e_i\}$ for $i = 1, \ldots, N^e$ where $N^e$ is the number of documents in $C^e$; while the documents in $C^j$ are noted as $\{d^j_i\}$ for $i = 1, \ldots, N^j$ where $N^j$ is the number of documents in $C^j$. Other than the data corpora, we also have a set of seed dictionaries. We assume that the seed dictionary comes from the generic domain, and is noisy. It means one
term in the seed dictionary may have several translations, within which some translations are not correct in this domain.

Now given a term in the source language \( t \) which appears in \( C \), we want to find the most possible translation term in \( C_e \).

### 3.2 Background of Cross-lingual Topic Models

As discussed in Section 2, topic models have been successfully used for lexicon extraction in parallel/comparable corpora. A classical bilingual LDA requires the documents to be aligned in pairs. The basic idea is that an aligned document pair should have the same topic distribution \( \theta \). For each document pair \( < d_{l1}, d_{l2} > \), a topic distribution \( \theta \) is drawn from a Dirichlet distribution:

\[
\theta \sim \text{Dirichlet}(\alpha)
\]

Then, for each language \( l \), a topic assignment is sampled for each word, as

\[
z_l \sim \text{Multinomial}(\theta)
\]

As the final step, words in each language are separately drawn from their topic assignment and topic-specific distribution

\[
\phi_{zl} \sim \text{Dir}(\beta_l)
\]

\[
w_l \sim \text{Multinomial}(\phi_{zl})
\]

In this way, the topics in different languages can be connected. Moreover, we can measure the similarity of documents in different languages, \( d_i, d_k \), simply by computing the similarity of their topic distributions \( \text{Sim}(\theta_i, \theta_k) \). The bilingual topic models can also be extended to multiple languages [13] and multiple modalities [2]. However, most of these models never consider the probability of the multiple translations or the noise in their dictionary. Boyd-Graber and Blei [4] integrate the prior of word matchings to the bilingual topic models in non-parallel data, but their model has no effect on finding new word translations.

The topic models for citation networks belong to another category related to our work [14, 6]. The idea is similar to the cross-lingual topic models. The cited document should have a similar topic distribution to the citing document. However, a document may have multiple cited documents, so its topic distribution is not totally the same as any of its cited documents. We will look at how these models motivate our new model later.

### 3.3 Model Description

#### 3.3.1 Inverted Indexing

Our approach to lexicon extraction is to first use topic models to model the cross-lingual data and obtain the topic distribution of each word. Then we can compare the topic distributions to compute the word similarities and get the translation.
In a conventional topic model, only the documents are represented by topic distributions, while the topic distribution for a word is not explicit. In addition, it is relatively easy to model document pairs or document relationships by various topic models, as discussed earlier. However, in our problem setting, we only have a seed dictionary and non-parallel data corpora, so it is difficult to find document relationships but easy to get word translation pairs. The motivation is that if we can transfer a word into a pseudo document, we can utilize the word relationship in seed dictionaries.

In order to implement this idea, we invert the document-word index so that a word is constructed by a list of document IDs. If we assume a word \( w \) that appears in \( d_1 \) twice, \( d_2 \) once, and \( d_3 \) once, it is represented as \((d_1, d_1, d_2, d_3)\). We also keep the word frequency in this representation.

As far as we know, this is the first work to integrate inverted indexing and topic models. Of course, there are other ways to construct the pseudo documents, such as using neighbor words. However, there are far fewer documents than context words, so we can reduce the computational cost. In addition, using inverted indexing-based representation enables us to easily calculate \( p(d|w) = \sum_z p(d|z)p(z|w) \) from the topic distributions. So we can easily achieve the conditional probability of all documents when given a search a term in another language. This might be useful for cross-lingual information retrieval tasks (although this is not our focus in this paper).

To avoid confusion, in the following sections we use "word" to refer to the pseudo document in topic models and use "document" to refer to the basic element in a pseudo document. Thus, a topic is a distribution of documents, and a word is a mixture of topics. That is to say, we have reversed the roles of words and documents in conventional topic models.

### 3.3.2 Inverted Bilingual LDA

Once we obtain the pseudo documents, we can use them to train a Bilingual LDA model. If two words are translations to each other, they are assumed to have similar topic distributions. The problem is that we only have a subset of words that are translated, and a word in a seed dictionary may have several translations. Therefore, first we need to construct one-to-one word pairs, the same as what Bilingual LDA does for documents.

Intuitively, it is not a good choice to make all translations modeled because if a word has polysemy, the different translations will own the same topic distribution. Instead, we just select the most frequent term in the translation list to form a translation pair. Then, for all translation pairs, we use the same model as the Bilingual LDA (Fig. 1). Words that do not have translations are modeled together using the original LDA.

- For each translation pair \( t^j, t^e \),
  - Sample a topic distribution \( \theta \sim \text{Dirichlet}(\alpha) \)
- For each word \( t_l \) (\( l \in \{j, e\} \)) without translation,
– Sample a topic distribution $\theta_i \sim \text{Dirichlet}(\alpha)$

Following this process, we sample the topics for each token $d^e$ and $d^j$ from theta (as in Section 3.2) and then draw documents from the topic. We also performed experiments to try out another way to obtain translation pairs. Instead of just selecting one translation, we randomly select a translation in each sample iteration, which means we finally use all the translations over all iterations. We call this model BiLDA_all, while the previous one is called BiLDA. A comparison of the two models is given in Section 5. They are both used as our baseline systems.

3.3.3 Probabilistically Linked Bilingual LDA

If we select just one translation, there is a risk of losing a lot of information. This is especially problematic when the seed dictionary is not large, as the lost information will cause a serious performance decrease. On the other hand, using all translations without discrimination is not ideal either, as we discussed previously. We therefore came up with a solution to properly select the correct translation for each word.

We developed two approaches to model the probability of translation selection. The first approach is to add a selection variable for each token (i.e. each document) $d^j$ in word $t^j$, such that the topic distribution of each $t^j$ is a mixture of its translations. This is similar to the idea of citation models, which model the probability of citation as the influence rate. The difference is that we have two sets of topics for the two respective languages. We do not directly share the topics of the "cited" pseudo document, opting instead to use the “cited”
Table 1: Notations for topic models in Figure 2, Figure 3, Figure 4

| Notation | Description |
|----------|-------------|
| $\alpha, \beta$ | hyperparameters for Dirichlet distribution |
| $\theta$ | topic distribution for a word |
| $\phi^e, \phi^j$ | document distribution for each topic |
| $z^e, z^j$ | topic assignment for each document |
| $d^e, d^j$ | documents in each word (i.e. IDs of the original documents that a word appears in.) |
| $\psi$ | distribution of the translation selections |
| $s$ | selecting a translation for a document (Figure 2) or for a word (Figure 3) |
| $M$ | number of words |
| $K$ | number of topics |

topic distribution to sample a new topic in its own language. We call this model **ProbBiLDA** (probabilistically linked bilingual LDA). The generative process of the ProbBiLDA is as follows. For a description of all the variables, please see Table 1.

- For each topic $z^l \in \{1,...,K\}$ in language $l$ ($l \in \{e,j\}$):
  - Sample document distribution $\phi^l \sim \text{Dir}(\beta)$

- For each word $t^e$:
  - Sample a topic distribution $\theta_{te} \sim \text{Dir}(\alpha)$
  - For each position $i$ in the word:
    * Sample a topic assignment from $z^e_i \sim \text{Multi}(\theta_{te})$
    * Draw a document $d^e_i \sim \text{Multi}(\phi^e_{z^e_i})$

- For each word $t^j$:
  - If this word does not have a translation in the seed dictionary:
    * Sample a topic distribution $\theta_{tj} \sim \text{Dir}(\alpha)$
    * For each position $i$ in the word:
      - Sample a topic assignment from $z^j_i \sim \text{Multi}(\theta_{tj})$
  - If the word has $S$ translations:
    * Draw a probability distribution $\psi_{tj} \sim \text{Dir}(\alpha_{\psi})$ over all translations
    * For each position $i$ in the word:
      - Sample a translation $s_i \sim \text{Multi}(\psi_{tj})$ from the $S$ translations
3.3.4 BlockProbBiLDA

Another way to model the probability of translations is to add the probability variable to the word itself instead of to each document in that word. That is to say, we select a translation for the whole word, and all the documents in that word follow the same topic distribution. For example, a word \( t = (d_1, d_2) \) has three translations \( t_1, t_2, t_3 \). If we use the ProbBiLDA, the topic of each document in word \( t \) is sampled from different translations, e.g., \( z_{d_1} \sim \theta_{t_2} \) and \( z_{d_2} \sim \theta_{t_3} \). However, in the new model, we require that all documents in \( t \) can only select one same translation in each iteration. If \( t_2 \) is selected as the translation of \( t \), we must have \( z_{d_1} \sim \theta_{t_2} \) and \( z_{d_2} \sim \theta_{t_2} \).

As all the documents select translations together like a block, we call this model **BlockProbBiLDA**. This model is essentially more similar to the original Bilingual LDA. Compared to Bilingual LDA, it does not fix the translation pairs but rather assigns a prior to each translation. Compared to the generative process of ProbBiLDA, it only changes the position of \( s \) and uses a uniform prior distribution \( \psi \) instead of Dirichlet prior. The graphical representation of BlockProbBiLDA is shown in Figure 3 and its generative process is as follows:

- For each topic \( z^l \in \{1, \ldots, K\} \) in language \( l \) \((l \in \{e, j\})\):
Figure 3: BlockProbBiLDA

- Sample document distribution $\phi^l \sim \text{Dir}(\beta)$

- For each word $t^e$:
  - Sample a topic distribution $\theta_{te} \sim \text{Dir}(\alpha)$ over the first $K$ topics
  - For each position $i$ in the pseudo document:
    * Sample a topic assignment from $z_{te}^i \sim \text{Multi}(\theta_{te})$
    * Draw a document $d_{te}^i \sim \text{Multi}(\phi^l_{te})$

- For each word $t^j$:
  - If this word does not have a translation in the seed dictionary:
    * Sample a topic distribution $\theta_{tj} \sim \text{Dir}(\alpha)$
    * For each position $i$ in the word:
      - Sample a topic assignment $z_{tj}^i \sim \text{Multi}(\theta_{tj})$
  - If the word has $S$ translations.
    * Sample a uniform probability distribution $\psi_{tj}$ over all translations
    * For each position $i$ in the pseudo document:
      - Sample a translation $s_i \sim \text{Multi}(\psi_{tj})$ from the $S$ translations.
      - Draw a topic $z_{tj}^i \sim \text{Multi}(\theta_{s_i})$
      - Draw a document $d_{tj}^i \sim \text{Multi}(\phi^l_{s_i})$
3.4 Posterior Inference

For both of the two new models, we use collapsed Gibbs sampling to approximate the posterior. We iteratively update latent variables (including topic assignment \(z\)) given other variables.

3.4.1 Posterior Inference for ProbBiLDA

In the model of ProbBiLDA, for each document \(d_{t,j,i}\) in a word \(t^j\), we assume that it selects a translation word \(c\) in target language \(e\), i.e. it is drawn from the topic distribution of this word. Given the translation selection, and other topic assignments, we sample the topic for document \(d_{t,j,i}\) according to:

\[
p(z_1^j = k | z_{-1, t,j}, s_i = c, d_{t,j,i} = n, \theta) \propto \frac{nmk(c, k) + cmk(c, k) + \alpha - 1}{nm(c) + cm(c) + K * \alpha - 1} \times \frac{nkj(k, n) + \beta - 1}{nkj(k) + V_\epsilon * \beta - 1}
\]

where \(nmk(c, k)\) denotes the number of documents in word \(c\) that are assigned to topic \(k\); \(cmk(c, k)\) denotes the number of documents with topic \(k\) in language \(e\) that select \(c\) as the translation of its associated word; and \(cm(c)\) is the total number of documents in language \(e\) with translation selection \(c\). \(nkj(k, n)\) is the number of times when document \(n\) is assigned to topic \(k\) in language \(j\); and accordingly \(nkj(k)\) is the sum of \(nkj(k, n)\) over all documents in language \(j\); \(V_\epsilon\) is the total number of documents in language \(j\).

Given these topic assignments, we can sample the translation selection:

\[
p(s_i = c | s_{-i}, z_1^j = k, d_{t,j,i} = n, \theta) \propto \prod_i \frac{nmk(t^i, c, k) + cmk(t^i, c, k) + \alpha - 1}{nm(t^i) + cm(t^i) + K * \alpha - 1} \times \frac{nms(t^i, c) + \alpha_\psi - 1}{nms(t^i) + S(t^i) * \alpha_\psi - 1}
\]

where \(nms(t^i, c)\) denotes the number of documents in word \(t^i\) which selects translation \(c\); \(nm(t^i)\) is the number of documents in word \(t^i\); and \(S(t^i)\) is the number of translation candidates for word \(t^i\).

The above sampling scheme is for the source language. While for target language, we only need to care about the topic assignments.

\[
p(z_1^e = k | z_{-1, t^e}, d_{t^e,i} = n, \theta) \propto \frac{nmk(t^e, k) + cmk(t^e, k) + \alpha - 1}{nm(t^e) + cm(t^e) + K * \alpha - 1} \times \frac{nkve(k, n) + \beta - 1}{nkve(k) + V_e * \beta - 1}
\]

where the denotations of the variables are similar to the ones defined in [1].

Given all the topic assignments, we can then derive the topic distribution \(\theta_m = (\theta_{m,1}, \theta_{m,2}, ..., \theta_{m,K})\) for word \(m\).

\[
\theta_{m,k} = \frac{nmk(m, k) + \alpha}{nm(m) + K * \alpha}
\]
The topic variables are derived from:

$$\phi_k^e = \frac{nkve(k, n) + \beta}{nke(k) + V_e \beta}$$  \hspace{1cm} (5)

$$\phi_k^j = \frac{nkvj(k, n) + \beta}{nkj(k) + V_j \beta}$$  \hspace{1cm} (6)

We run 1500 iterations for inference while the first 1000 iterations are discarded as burn-in steps. After the sampling chain converges, we average the value of \(\theta_m\) to get the final per-word topic distribution.

### 3.4.2 Posterior Inference for BlockProbBiLDA

For each word \(t^e\), we sample its topic according to:

$$p(z^e_i = k | z^e_{-i,t^e}, d^e_{-i,t^e} = n, \theta)$$ \hspace{1cm} (7)

$$\propto \frac{nmk(t^e,k) + cmk(t^e,k) + \alpha - 1}{nm(t^e) + cm(t^e) + K * \alpha - 1} \cdot \frac{nkve(k, n) + \beta - 1}{nke(k) + V_e \beta - 1}$$

For each word \(t^j\), if it is in the dictionary, and it selects \(c\) as its translation in the previous iteration, then

$$p(z^j_i = k | z^j_{-i,t^j}, d^j_{-i,t^j} = d, \theta)$$ \hspace{1cm} (8)

$$\propto \frac{nmk(t^j,k) + nkj(k) + \alpha - 1}{nm(t^j) + nkj(k) + V_j \beta - 1}$$

The selection of translations is sampled by

$$p(s^j = t^e | z^j_i, d^j_{-i,t^j} = d, \theta)$$ \hspace{1cm} (9)

$$\propto \prod_i \frac{nmk(t^e, z^j_i) + \alpha + \sum_{m \in C(t^e)/\{t^j\}} nmk(m, z^j_i)}{nm(t^e) + \alpha + \sum_{m \in C(t^e)/\{t^j\}} nm(m)}$$

where \(C(t^e)\) is the set of all words which cite \(t^e\) as their translations in last iteration; \(C(t^e)/\{t^j\}\) means to exclude \(t^j\) in this set. As the product of the probabilities is usually very small, \(p(s^j = t^e)\) has different orders of magnitude for each \(t^e\), so the sampling of \(s^j\) can be approximated by selecting the one with largest probability. We use the following equation instead:

$$s^j \approx \arg \max_{t^e} \sum_i \log \frac{nmk(t^e, z^j_i) + \alpha + \sum_{m \in C(t^e)/\{t^j\}} nmk(m, z^j_i)}{nm(t^e) + \alpha + \sum_{m \in C(t^e)/\{t^j\}} nm(m)}$$ \hspace{1cm} (10)

After sampling the translation selection \(s^j = t^e\) for \(t^j\), we update the \(C(t^e)\) as well as \(C(t^j)\), where \(C(t^j)\) is the previous selection of \(s^j\). Then we use a scheme similar to the one used in 3.4.1 to get topic distribution \(\theta\).
4 Computing Word Similarities to Obtain Translations

Once we get the topic distribution of each word, we can use them to calculate the similarity between words. The simplest way to do this is to regard each topic distribution as a vector representation of a word. We can then calculate the cosine similarity between these vectors.

\[
\text{Cosine}(\theta_m, \theta_c) = \frac{\sum_{k=1}^{K} \theta_{mk} \theta_{ck}}{\sqrt{\sum_{k=1}^{K} \theta_{mk}^2} \sqrt{\sum_{k=1}^{K} \theta_{ck}^2}}
\]

(11)

Another measure is to use the Kullback-Leibler (KL) divergence. KL divergence is a measure of difference between two probability distributions that is widely used in previous topic model-based approaches.

\[
D_{KL}(\theta_m || \theta_c) = \sum_{k=1}^{K} \theta_{mk} \log \frac{\theta_{mk}}{\theta_{ck}}
\]

(12)

Neither cosine similarity nor KL divergence considers the correlation between topics. For a topic model, as we know the topic distribution of each word in addition to knowing the topic itself, we can take advantage of the topic structures by directly modeling the probability of \( p(w^e|w^j) \) as the similarity between words \( w^e \) and \( w^j \). This tells us how likely it is we can generate \( w^e \) from \( w^j \). We call this similarity measure selProb (selection probability).

\[
\text{selProb} = p(w^e|w^j) \propto p(w^j|\theta_{w^e})
\]

\[
= \prod_{i=1}^{n} \sum_{z^j=1}^{K} p(d_i^j|z^j, \phi^j) p(z^j|\theta_{w^e})
\]

Then, we can select the most similar word in the target language as the translation.

\[
\arg \max_{w^e} \log p(w^e|\theta_{w^e})
\]

\[
= \arg \max_{w^e} \sum_{i=1}^{n} \log \sum_{z^j=1}^{K} p(d_i^j|z^j, \phi^j) p(z^j|\theta_{w^e})
\]

5 Experimental Results

5.1 Experiment Data

We use two Japanese-English domain-specific corpora for our experiments. The first corpus comes from a bilingual law dataset \[^3\]. We selected it for our experiments because it has an associated law dictionary that can be directly used

\[^3\]http://www.phontron.com/jaen-law/index-ja.html
Table 2: Accuracies of new words on law data.

| Method                  | Acc1 | Acc10 |
|-------------------------|------|-------|
| TFIDF                   | 5.2% | 20.3% |
| LP                      | 3.8% | 15.0% |
| mixed word embedding    | 1.5% | 15.0% |
| BiLDA + cosine          | 4.5% | 9.8%  |
| BiLDA + KLD             | 6.8% | 18.0% |
| BiLDA + selProb         | 6.0% | 17.3% |
| BiLDA_all + cosine      | 1.5% | 9.0%  |
| BiLDA_all + KLD         | 3.0% | 10.5% |
| BiLDA_all + selProb     | 2.25%| 12.0% |
| ProbBiLDA + cosine      | 9.8% | 17.3% |
| ProbBiLDA + KLD         | 5.3% | 18.0% |
| ProbBiLDA + selProb     | 11.3%| 24.1% |
| BlockProbBiLDA + cosine | 9.0% | 23.3% |
| BlockProbBiLDA + KLD    | 14.3%| 29.3% |
| BlockProbBiLDA + selProb| 14.3%| 31.6% |

as test data. Although this corpus was originally parallel, we do not use it as a parallel data source. We randomly select 150,000 paragraphs for each language (each paragraph is seen as one document) and the paragraphs are not kept aligned. The other dataset is a collection of car complaints from MLIT⁴ and NHTSA⁵. The data, which is unbalanced, includes 351,811 short English documents and 32,059 short Japanese documents.

The Japanese texts are processed by our own NLP tool to obtain the segmentation and the English texts are tokenized and lemmatized by NLTK⁶. We removed stop words for both Japanese and English texts. We also removed documents containing less than five words. For both corpora, we use the generic Japanese-English dictionary Edict⁷ and exclude any words that do not appear in our corpora. For the law data, we use the associated law dictionary as the test data. After erasing any words not in the corpora, we are left with 840 words for test data, within which 133 words are not in the generic dictionary and 254 do not have correct translations in the generic dictionary. In terms of the car data, we do not have any technical dictionary for cars, so we manually annotate 150 technical words not covered by the seed dictionary. In contrast to [1] and [19], we test not only nouns but also words of other parts of speech.

5.2 Comparisons with other models

We compare our approaches with several previous approaches.

⁴http://www.mlit.go.jp/jidosha/carinf/rcl/defects.html
⁵http://www-odi.nhtsa.dot.gov/downloads/index.cfm
⁶http://www.nltk.org
⁷http://www.edrdg.org/jmdict/edict.html
Table 3: Accuracies of all test words on Law Data.

| Method                        | Acc1  | Acc10 |
|-------------------------------|-------|-------|
| TFIDF                         | 57.6% | 72.1% |
| LP                            | 44.2% | 71.7% |
| Mixed word embedding          | 43.8% | 71.9% |
| BiLDA + cosine                | 56.3% | 73.4% |
| BiLDA + KLD                   | 58.2% | 74.8% |
| BiLDA + selProb               | 53.3% | 74.9% |
| BiLDA_all + cosine            | 56.3% | 72.6% |
| BiLDA_all + KLD               | 56.4% | 72.7% |
| BiLDA_all + selProb           | 52.1% | 73.8% |
| ProbBiLDA + cosine            | 55.7% | 74.7% |
| ProbBiLDA + KLD               | 54.8% | 75.2% |
| ProbBiLDA + selProb           | 57.4% | 77.4% |
| BlockProbBiLDA + cosine       | 58.1% | 76.1% |
| BlockProbBiLDA + KLD          | 59.3% | 77.6% |
| BlockProbBiLDA + selProb      | 60.5% | 78.1% |

- **TFIDF** [7] is a classical lexicon extraction method. It uses tf-idf weights for contextual seed words to obtain the context vectors and then uses cosine similarity to rank candidate translations.

- **Label Propagation (LP)** propagates seed distributions on a graph representing relations among words, and translation pairs are extracted by identifying word pairs with a high similarity in the seed distributions. It is claimed that it could resolve the sparseness problem of context vectors and better utilize the information of seed dictionary. In our experiments, we use the same parameter setting as [19] (window size = 4).

- **Mixed word embedding** [8] is a simple but efficient cross-lingual word embedding that can handle the problem of multiple translation (unlike the one-to-one seed dictionary used in [12]). After obtaining the word embeddings, we use cosine similarity to rank the candidates.

Two Bilingual LDAs (see 3.3.2) are implemented as our baselines. The ProbBiLDA and BlockProbBiLDA are our final models. For all topic models, we use three similarity measures to rank the candidate translations: cosine similarity, KL divergence, and selProb. We use Accuracy@K (K=1, 10) as the evaluation metric. Note that for law data, we test not only the overall accuracy of all law dictionary words (Acc1_full, Acc10_full) but also the accuracy w.r.t only new words (Acc1_new, Acc10_new). As for car data, all test words are new, so we do not discriminate them. The results are listed in Table 2, Table 3 and Table 4.

First let us compare the three similarity measures. It is obvious that selProb performs better than the other two in most cases. This supports our assumption
Table 4: Performance on Car Data

| Method                        | Acc1 | Acc10 |
|-------------------------------|------|-------|
| TFIDF                         | 1.3% | 3.3%  |
| LP                            | 2.7% | 7.3%  |
| Mixed word embedding          | 4.7% | 8.7%  |
| BiLDA + cosine                | 4%   | 12%   |
| BiLDA + KLD                   | 4.7% | 9.3%  |
| BiLDA + selProb               | 5.3% | 12.7% |
| BiLDA_all + cosine            | 4.7% | 9.3%  |
| BiLDA_all + KLD               | 4%   | 6.7%  |
| BiLDA_all + selProb           | 6%   | 13.3% |
| ProbBiLDA + cosine            | 2.7% | 7.3%  |
| ProbBiLDA + KLD               | 2.7% | 4.0%  |
| ProbBiLDA + selProb           | 6%   | 11.3% |
| BlockProbBiLDA + cosine       | 6.7% | 10%   |
| BlockProbBiLDA + KLD          | 4.0% | 8.0%  |
| BlockProbBiLDA + selProb      | 8.0% | 14.6% |

that topics are correlated and selProb can better catch the word similarity. In
the following comparison, we use only this measure for all topic models.

By comparing LP and TFIDF, we find, somewhat surprisingly, that LP per-
forms worse than TFIDF on law data (Table 2 and Table 3). This is probably
because the seed dictionary is very noisy for the law domain. Because of the
noise, LP does not benefit from propagating the seed dictionary—on the con-
trary, it is badly impacted because error translations are propagated. We may
achieve a similar conclusion by comparing BiLDA and BiLDA_all. However, on
car data (Table 4), the comparison result is reversed, so we can only assume
that the seed dictionary for the car domain is cleaner.

Now let us turn to our two probabilistically linked bilingual topic models,
the ProbBiLDA and BlockProbBiLDA. Regardless of the datasets, BlockProb-
BiLDA is consistently the best model and the ProbBiLDA is the runner-up or
at least comparable to other best models. This demonstrates the effectiveness
of modeling translation probability. Among the two models, BlockProbBiLDA is
better on both datasets. A possible explanation is that BlockProbBiLDA tends
to exclude the noise when a word has several translations, while ProbBiLDA is a
mixed average of all translations including noise; and BlockProbBiLDA enables
words better correlated to their translations than ProbBiLDA.

5.3 Examples of Extracted Translation Pairs

Table 5 shows the examples of translations for several typical law terminologies.
In the table, 申立, 質権 and 源泉徵収 are words that do not have translations
in the seed dictionary; while 更生 has multiple translations (including "rehabil-
itation", "reorganization", "reuse" and "recycle") in the seed dictionary. For
each word, we select 10 highest ranked candidates. We can see that most of the translation candidates for the new words (which are not included in the seed dictionary) are semantically related. And for a polyseme, we can easily get the correct meaning in the special domain. Thus given a new corpus, our technique is useful to help people refine the generic dictionary for a special domain; and we could also expand the generic dictionary by extracting new translation pairs in the corpus. In addition, we may also find semantically related words which help us understand the words.

Table 5: Some examples of the extracted translation pairs. We use bold fonts to denote the correct translations.

| 申立書 | 質権 | 更生 | 源泉徴収 |
|--------|------|------|----------|
| petition | pledge | rehabilitation | attribute |
| proceeding | pledgee | reorganization | taxation |
| merit | pldgees | reuse | affiliate |
| clerk | perfection | recycle | domestic |
| petitioner | holder | restrain | withholding |
| court | co-owner | compulsary | sum |
| prejudice | entitlement | bankruptcy | resident |
| pend | division | avoidance | withhold |
| deny | recording | guardianship | nonpermanent |
| stand | convertible | remedy | dividend |

5.4 Discussion and Future Work

Someone may argue that in cross-lingual datasets, especially un-aligned corpora, there should be language-specific topics that are not shared across languages. Intuitively we can build a more general BlockProbBiLDA model combined with possible language-specific topics. However, our experiments show that this idea does not work well, and it does not improve BlockProbBiLDA at all. So we do not use this more general model in this paper.

Nonetheless, we believe that there is still space to improve our models. Our future work might include using different representation (e.g. inverted indexing and context vectors) of words in our models and integrate the different results. We may also apply our models to cross-lingual information retrieval or other related tasks.

6 Conclusion

In this paper, we proposed a new framework for extracting translations from non-parallel corpora. First we constructed pseudo documents using inverted indexing; and then we introduced two new bilingual topic models, ProbBiLDA and BlockProbBiLDA, to obtain topic distributions for each word. These models are extensions of the classical Bilingual LDA featuring a new hierarchy to
integrate the translation probability for multiple translations in the seed dictionary. We advanced the generation of probability to measure the similarity between one candidate word and a given target word. Experimental results show that Bilingual Topic models as well as inverted representation can be effectively utilized for lexicon extraction and that modeling the probabilities of translations in the seed dictionary is helpful as well.

References

[1] D. Andrade, T. Nasukawa, and J. Tsujii. Robust measurement and comparison of context similarity for finding translation pairs. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 19–27. Association for Computational Linguistics, 2010.

[2] D. M. Blei and M. I. Jordan. Modeling annotated data. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pages 127–134. ACM, 2003.

[3] D. Bollegala, G. Kontonatsios, and S. Ananiadou. A cross-lingual similarity measure for detecting biomedical term translations. PloS one, 10(6):e0126196, 2015.

[4] J. Boyd-Graber and D. M. Blei. Multilingual topic models for unaligned text. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pages 75–82. AUAI Press, 2009.

[5] J. Chang and D. M. Blei. Relational topic models for document networks. In International conference on artificial intelligence and statistics, pages 81–88, 2009.

[6] L. Dietz, S. Bickel, and T. Scheffer. Unsupervised prediction of citation influences. In Proceedings of the 24th international conference on Machine learning, pages 233–240. ACM, 2007.

[7] P. Fung and L. Y. Yee. An ir approach for translating new words from nonparallel, comparable texts. In Proceedings of the 17th international conference on Computational linguistics-Volume 1, pages 414–420. Association for Computational Linguistics, 1998.

[8] S. Gouws and A. Søgaard. Simple task-specific bilingual word embeddings. In Proceedings of NAACL-HLT, pages 1386–1390, 2015.

[9] A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. Learning bilingual lexicons from monolingual corpora. ACL-08: HLT, page 771, 2008.

[10] F. Laws, L. Michelbacher, B. Dorow, C. Scheible, U. Heid, and H. Schütze. A linguistically grounded graph model for bilingual lexicon extraction. In
Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 614–622. Association for Computational Linguistics, 2010.

[11] A. Lopez. Statistical machine translation. ACM Computing Surveys (CSUR), 40(3):8, 2008.

[12] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168, 2013.

[13] D. Mimno, H. M. Wallach, J. Naradowsky, D. A. Smith, and A. McCallum. Polyligual topic models. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2, pages 880–889. Association for Computational Linguistics, 2009.

[14] R. Nallapati and W. W. Cohen. Link-plsa-lda: A new unsupervised model for topics and influence of blogs. In ICWSM, 2008.

[15] V. Pekar, R. Mitkov, D. Blagoev, and A. Mulloni. Finding translations for low-frequency words in comparable corpora. Machine Translation, 20(4):247–266, 2006.

[16] R. Rapp. Identifying word translations in non-parallel texts. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics, pages 320–322. Association for Computational Linguistics, 1995.

[17] D. Shezaf and A. Rappoport. Bilingual lexicon generation using non-aligned signatures. In Proceedings of the 48th annual meeting of the association for computational linguistics, pages 98–107. Association for Computational Linguistics, 2010.

[18] A. Søgaard, Ž. Agić, H. M. Alonso, B. Plank, B. Bohnet, and A. Johannsen. Inverted indexing for cross-lingual nlp. In The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015), 2015.

[19] A. Tamura, T. Watanabe, and E. Sumita. Bilingual lexicon extraction from comparable corpora using label propagation. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 24–36. Association for Computational Linguistics, 2012.

[20] I. Vulić, W. De Smet, and M.-F. Moens. Identifying word translations from comparable corpora using latent topic models. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, pages 479–484. Association for Computational Linguistics, 2011.
[21] I. Vulic and M.-F. Moens. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015). ACL, 2015.