Iterative Learning of Parallel Lexicons and Phrases from Non-Parallel Corpora

Meiping Dong[†] **Yang Liu**[†‡] [*] **Huanbo Luan**[†] **Maosong Sun**[†‡] **Tatsuya Izuha**[+] **Dakun Zhang**[#]
[†]State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Sci. and Tech., Tsinghua University, Beijing, China
[‡]Jiangsu Collaborative Innovation Center for Language Competence, Jiangsu, China
hellodmp@163.com, {liuyang2011,sms}@tsinghua.edu.cn, luanhuanbo@gmail.com
[+]Toshiba Corporation Corporate Research & Development Center
tatsuya.izuha@toshiba.co.jp
[#]Toshiba (China) R&D Center
zhangdakun@toshiba.com.cn

Abstract

While parallel corpora are an indispensable resource for data-driven multilingual natural language processing tasks such as machine translation, they are limited in quantity, quality and coverage. As a result, learning translation models from non-parallel corpora has become increasingly important nowadays, especially for low-resource languages. In this work, we propose a joint model for iteratively learning parallel lexicons and phrases from non-parallel corpora. The model is trained using a Viterbi EM algorithm that alternates between constructing parallel phrases using lexicons and updating lexicons based on the constructed parallel phrases. Experiments on Chinese-English datasets show that our approach learns better parallel lexicons and phrases and improves translation performance significantly.

1 Introduction

Parallel corpora, which are collections of parallel texts, play a critical role in data-driven multilingual natural language processing (NLP) tasks such as statistical machine translation (MT) and cross-lingual information retrieval. For example, in statistical MT, parallel corpora serve as the central source for estimating translation model parameters [Brown *et al.*, 1993; Koehn *et al.*, 2003; Chiang, 2005]. It is widely accepted that the quantity, quality, and coverage of parallel corpora have an important effect on the performance of statistical MT systems.

Despite the apparent success of data-driven multilingual NLP techniques, the availability of large-scale, wide-coverage, high-quality parallel corpora still remains a major challenge. For most language pairs, parallel corpora are nonexistent. Even for the top handful of resource-rich languages, the available parallel corpora are usually unbalanced because the major sources are government documents or news articles.

As a result, learning translation models from non-parallel corpora has attracted intensive attention from the community [Koehn and Knight, 2002; Fung and Cheung, 2004; Munteanu and Marcu, 2006; Quirk *et al.*, 2007; Ueffing *et al.*, 2007; Haghighi *et al.*, 2008; Bertoldi and Federico, 2009; Cettolo *et al.*, 2010; Daumé III and Jagarlamudi, 2011; Ravi and Knight, 2011; Nuhn *et al.*, 2012; Dou and Knight, 2012; Klementiev *et al.*, 2012; Zhang and Zong, 2013; Dou *et al.*, 2014]. Most existing approaches focus on learning *word-based* models: either bilingual lexicons or IBM models. Based on canonical correlation analysis (CCA), Haghighi et al. [2008] leverage orthographic and context features to induce word translation pairs. Ravi and Knight [2011] cast training IBM models on monolingual data as a decipherment problem. However, word-based models are not expressive enough to capture non-local dependencies and therefore are insufficient to yield high quality translations.

Recently, several authors have moved a step further to learn *phrase-based* models from non-parallel corpora [Klementiev *et al.*, 2012; Zhang and Zong, 2013]. Zhang and Zong [2013] propose to use a parallel lexicon to retrieve parallel phrases from non-parallel corpora. They show that their approach can learn new translations and improve translation performance. However, their approach is unidirectional: only using lexicons to identify parallel phrases. In fact, it is possible to learn better lexicons from extracted phrase pairs in a reverse direction, which potentially constitutes a "*find-one-get-more*" loop [Fung and Cheung, 2004].

In this paper, we propose an iterative approach to learning bilingual lexicons and phrases jointly from non-parallel corpora. Given two sets of monolingual phrases that might contain parallel phrases, we develop a generative model based on IBM model 1 [Brown *et al.*, 1993], which treats the mapping between phrase pairs as a latent variable. The model is trained using a Viterbi EM algorithm. Experiments on Chinese-English datasets show that iterative training significantly improves the quality of learned bilingual lexicons and phrases and benefit end-to-end MT systems.

2 Preliminaries

We begin with a brief introduction to IBM model 1, which is the core component of our generative model.

Let $\mathbf{f} = \mathbf{f}_1, \ldots, \mathbf{f}_J$ be a foreign sentence with J words. We use $\mathbf{f}_j$ to denote the j-th word in the foreign sentence. Similarly, $\mathbf{e} = \mathbf{e}_1, \ldots, \mathbf{e}_I$ is an English sentence with I words and $\mathbf{e}_i$ is the i-th word. f and e denote single foreign and English words, respectively. A word alignment $\mathbf{a} = \mathbf{a}_1, \ldots, \mathbf{a}_J$

[*]Yang Liu is the corresponding author.

indicates the correspondence between $\mathbf{f}$ and $\mathbf{e}$. Brown et al. [1993] restrict that each foreign word is aligned to exactly one English word: $\mathbf{a}_j \in \{0, 1, \ldots, I\}$, where $\mathbf{e}_0$ is defined to be an empty English word.

According to Brown et al. [1993], IBM model 1 is defined as follows:

$$P(\mathbf{f}|\mathbf{e};\boldsymbol{\theta}) = \sum_{\mathbf{a}} P(\mathbf{f},\mathbf{a}|\mathbf{e};\boldsymbol{\theta}) \quad (1)$$

$$= \sum_{\mathbf{a}} \frac{p(J|I)}{(I+1)^J} \prod_{j=1}^{J} p(\mathbf{f}_j|\mathbf{e}_{\mathbf{a}_j}) \quad (2)$$

where the set of model parameters $\boldsymbol{\theta}$ consists of *length probabilities* $p(J|I)$ and *translation probabilities* $p(f|e)$:

$$\forall I : \sum_{J} p(J|I) = 1 \quad (3)$$

$$\forall e : \sum_{f} p(f|e) = 1 \quad (4)$$

Brown et al. [1993] indicate that the right-hand side of Eq. (2) is a sum of terms each of which is a monomial in the translation probabilities. Therefore, we have

$$P(\mathbf{f}|\mathbf{e};\boldsymbol{\theta}) = \frac{p(J|I)}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} p(\mathbf{f}_j|\mathbf{e}_i) \quad (5)$$

Given a set of parallel sentences $D = \{\langle \mathbf{f}^{(s)}, \mathbf{e}^{(s)} \rangle\}_{s=1}^{S}$, the training objective is to maximize the log-likelihood of the training data: [1]

$$J(\boldsymbol{\theta}) = \sum_{s=1}^{J} \log P(\mathbf{f}^{(s)}|\mathbf{e}^{(s)}) - \sum_{e} \lambda_e \Big(\sum_{f} p(f|e) - 1 \Big) \quad (6)$$

Brown et al. [1993] use the EM algorithm to estimate translation probabilities. $c(f|e;D)$, which is the expected number of times that e connects to f on the training data, is calculated as

$$\sum_{s=1}^{S} \frac{p(f|e)}{\sum_{i=0}^{I^{(s)}} p(f|\mathbf{e}_i^{(s)})} \sum_{j=1}^{J^{(s)}} \delta(f, \mathbf{f}_j^{(s)}) \sum_{i=0}^{I^{(s)}} \delta(e, \mathbf{e}_i^{(s)}) \quad (7)$$

where $I^{(s)}$ and $J^{(t)}$ are lengths of $\mathbf{e}^{(s)}$ and $\mathbf{f}^{(t)}$, respectively.

Then, the new translation probabilities can be obtained by normalizing these counts:

$$p(f|e) = \frac{c(f|e;D)}{\sum_{f'} c(f'|e;D)} \quad (8)$$

3 Model

In this work, we are interested in learning IBM Model 1 from non-parallel corpora: a set of foreign strings $F = \{\mathbf{f}^{(t)}\}_{t=1}^{T}$ and a set of English strings $E = \{\mathbf{e}^{(s)}\}_{s=1}^{S}$. Our goal is twofold:

1. Extract a parallel corpus from non-parallel corpora F and E using IBM model 1,
2. Train IBM model 1 from the extracted parallel corpus.

Due to data sparsity, a long foreign sentence in F can hardly have an English translation in E. Alternatively, we assume that a short foreign string, say a phrase with up to 7 words, in F can potentially have an English translation in E [Munteanu and Marcu, 2006; Quirk *et al.*, 2007; Cettolo *et al.*, 2010]. Therefore, in this work, F and E are defined to be sets of phrases. We can easily obtain such monolingual phrase sets by collecting n-grams.

We introduce *phrase matching* to denote the correspondence between phrases in F and E. A foreign phrase $\mathbf{f}$ is said to *match* an English phrase $\mathbf{e}$ if they are translations of each other. More formally, we use M to denote a set of all possible matchings and $\boldsymbol{m}$ a matching. Following Brown et al. [1993], we restrict that each foreign phrase matches exactly one English phrase: $\mathbf{m} = \mathbf{m}_1, \ldots, \mathbf{m}_T$, where $\mathbf{m}_t \in \{0, 1, \ldots, S\}$. Note that we allow a foreign phrase to connect to an empty English phrase $\mathbf{e}^{(0)}$.

The joint model is defined as

$$P(F|E;\boldsymbol{\theta}) = \sum_{\mathbf{m}} P(F,\mathbf{m}|E;\boldsymbol{\theta}) \quad (9)$$

$$= \sum_{\mathbf{m}} \frac{p(T|S)}{(S+1)^T} \prod_{t=1}^{T} P(\mathbf{f}^{(t)}|\mathbf{e}^{(\mathbf{m}_t)};\boldsymbol{\theta}) \quad (10)$$

We refer to $P(\mathbf{f}^{(t)}|\mathbf{e}^{(\mathbf{m}_t)};\boldsymbol{\theta})$ as *phrase translation* model, which can be divided into two categories: *empty* and *non-empty*. The probability of a foreign phrase given an empty English phrase is defined to be a model parameter satisfying

$$\sum_{\mathbf{f} \in \mathcal{F}} p(\mathbf{f}|\mathbf{e}^{(0)}) = 1 \quad (11)$$

where $\mathcal{F}$ is the set of all possible foreign phrases. Note that F in the training data is just a subset of $\mathcal{F}$. As $\mathcal{F}$ cannot be fully enumerated, we set $p(\mathbf{f}|\mathbf{e}^{(0)})$ to a fixed number ϵ, which is a hyper-parameter to be optimized on the held-out data.

For non-empty English phrases, we use IBM Model 1 as follows:

$$P(\mathbf{f}^{(t)}|\mathbf{e}^{(\mathbf{m}_t)};\boldsymbol{\theta}) = \frac{p(J^{(t)}|I^{(\mathbf{m}_t)})}{(I^{(\mathbf{m}_t)}+1)^{J^{(t)}}} \prod_{j=1}^{J^{(t)}} \sum_{i=0}^{I^{(\mathbf{m}_t)}} p(\mathbf{f}_j^{(t)}|\mathbf{e}_i^{(\mathbf{m}_t)}) \quad (12)$$

Note that the length model $p(J^{(t)}|I^{(\mathbf{m}_t)})$ plays an important role in learning IBM model 1 from non-parallel corpora since the model needs to choose among source phrases with various lengths.

Therefore, the phrase translation model is defined as

$$\begin{aligned} & P(\mathbf{f}^{(t)}|\mathbf{e}^{(\mathbf{m}_t)};\boldsymbol{\theta}) \\ = \; & \delta(\mathbf{m}_t, 0)\epsilon + \\ & (1 - \delta(\mathbf{m}_t, 0)) \frac{p(J^{(t)}|I^{(\mathbf{m}_t)})}{(I^{(\mathbf{m}_t)}+1)^{J^{(t)}}} \prod_{j=1}^{J^{(t)}} \sum_{i=0}^{I^{(\mathbf{m}_t)}} p(\mathbf{f}_j^{(t)}|\mathbf{e}_i^{(\mathbf{m}_t)}) \end{aligned} \quad (13)$$

[1] As there is no need to train the length model on parallel corpora, Brown et al. [1993] set $p(J|I)$ to some small, fixed number.

As learning translation models from non-parallel corpora is a very challenging task, many authors assume that a small parallel lexicon is readily available [Zhang and Zong, 2013]. Hence, we define a penalization term similar to the Kullback-Leibler divergence to incorporate prior knowledge into the model:

$$\sum_f \sum_e \sigma(f, e, \mathbf{d}) \log \frac{\sigma(f, e, \mathbf{d})}{p(f|e)} \tag{14}$$

where $\mathbf{d}$ is a parallel lexicon and $\sigma(f, e, \mathbf{d})$ checks whether $\langle f, e \rangle$ exists in the lexicon:

$$\sigma(f, e, \mathbf{d}) = \begin{cases} 1 & \text{if } \langle f, e \rangle \in \mathbf{d} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

4 Training

4.1 The Viterbi EM Algorithm

Given monolingual corpora F and E and a parallel lexicon $\mathbf{d}$, the training objective is given by

$$\begin{aligned} J(\boldsymbol{\theta}) &= P(F|E;\boldsymbol{\theta}) - \\ &\quad \sum_f \sum_e \sigma(f, e, \mathbf{d}) \log \frac{\sigma(f, e, \mathbf{d})}{p(f|e)} - \\ &\quad - \sum_I \lambda_I \Big(\sum_J p(J|I) - 1 \Big) - \\ &\quad - \sum_e \gamma_e \Big(\sum_f p(f|e) - 1 \Big) \end{aligned} \tag{16}$$

It is natural to use the EM algorithm to estimate the model parameters. The expected count of the length model $c(J|I; F, E)$ is given by

$$\mathbb{E}_{F,\mathbf{m}|E;\boldsymbol{\theta}} \Big[\sum_{t=1}^{T} (1 - \delta(\mathbf{m}_t, 0)) \delta(J^{(t)}, J) \delta(I^{(\mathbf{m}_t)}, I) \Big] \tag{17}$$

The expected count of the translation model $c(f|e; F, E)$ is given by

$$\begin{aligned} &\mathbb{E}_{F,\mathbf{m}|E;\boldsymbol{\theta}} \Big[\sum_{t=1}^{T} (1 - \delta(\mathbf{m}_t, 0)) \frac{p(f|e)}{\sum_{i=0}^{I^{(\mathbf{m}_t)}} p(f|\mathbf{e}_i^{(\mathbf{m}_t)})} \\ &\qquad \times \sum_{j=1}^{J^{(t)}} \delta(f, \mathbf{f}_j^{(t)}) \sum_{i=0}^{I^{(\mathbf{m}_t)}} \delta(e, \mathbf{e}_i^{(\mathbf{m}_t)}) \Big] \\ &+ \sigma(f, e, \mathbf{d}) \end{aligned} \tag{18}$$

Unfortunately, calculating the two expectations in Eq. (17) and (18) on the training data is intractable due to the exponential search space of phrase matching.

Instead, we use a Viterbi EM algorithm as shown in Figure 1. The algorithm takes a set of foreign phrases F, a set of English phrases E, and a parallel lexicon $\mathbf{d}$ as input (line 1). After initializing model parameters (line 2), the algorithm calls the procedure $\textsc{Align}(F, E, \boldsymbol{\theta})$ to compute the *Viterbi matching* between F and E (line 4). Then, the algorithm updates the model by normalizing counts collected from the Viterbi matching (line 5). This process terminates after K iterations and returns the final matching and model.

1: **procedure** VITERBIEM($F, E, \mathbf{d}$)
2: Initialize $\boldsymbol{\theta}^{(0)}$
3: **for all** $k = 1, \ldots, K$ **do**
4: $\hat{\mathbf{m}}^{(k)} \leftarrow \textsc{Align}(F, E, \boldsymbol{\theta}^{(k-1)})$
5: $\boldsymbol{\theta}^{(k)} \leftarrow \textsc{Update}(F, E, \mathbf{d}, \hat{\mathbf{m}}^{(k)})$
6: **end for**
7: **return** $\hat{\mathbf{m}}^{(K)}, \boldsymbol{\theta}^{(K)}$
8: **end procedure**

Figure 1: A Viterbi EM algorithm for learning IBM model 1 from non-parallel corpora. F and E are sets of foreign and English phrases, $\boldsymbol{\theta}^{(k)}$ is the set of parameters at the k-th iteration, and $\hat{\mathbf{m}}^{(k)}$ is the Viterbi matching between F and E.

4.2 Computing Viterbi Matching

Given a set of foreign phrases F and a set of English phrases E, the Viterbi matching is defined as

$$\hat{\mathbf{m}} = \operatorname*{argmax}_{\mathbf{m}} \Big\{ P(F, \mathbf{m}|E; \boldsymbol{\theta}) \Big\} \tag{19}$$

$$= \operatorname*{argmax}_{\mathbf{m}} \Big\{ \prod_{t=1}^{T} P(\mathbf{f}^{(t)}|\mathbf{e}^{(\mathbf{m}_t)}; \boldsymbol{\theta}) \Big\} \tag{20}$$

It is clear that computing the Viterbi matching for individual foreign phrases is *independent*. We only need to focus on finding the most probable English phrase for each foreign phrase:

$$\hat{\mathbf{m}}_t = \operatorname*{argmax}_{s \in \{0,1,\ldots,S\}} \Big\{ P(\mathbf{f}^{(t)}|\mathbf{e}^{(s)}; \boldsymbol{\theta}) \Big\} \tag{21}$$

For the empty English phrase (i.e., $s = 0$), the translation probability is simply ϵ. For non-empty English phrases, the decision rule is

$$\tilde{\mathbf{m}}_t = \operatorname*{argmax}_{s \in \{1,\ldots,S\}} \Big\{ \frac{p(J^{(t)}|I^{(s)})}{(I^{(s)} + 1)^{J^{(t)}}} \prod_{j=1}^{J^{(t)}} \sum_{i=0}^{I^{(s)}} p(\mathbf{f}_j^{(t)}|\mathbf{e}_i^{(s)}) \Big\} \tag{22}$$

Therefore, the Viterbi matching for a foreign phrase can be determined by

$$\hat{\mathbf{m}}_t = \begin{cases} \tilde{\mathbf{m}}_t & \text{if } P(\mathbf{f}^{(t)}|\mathbf{e}^{(\tilde{\mathbf{m}}_t)}; \boldsymbol{\theta}) > \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

We can see that ϵ determines whether a foreign phrase is unaligned or not. This is important for preventing a foreign phrase that has no counterparts on the English side from connecting to a wrong English phrase.

Computing $\tilde{\mathbf{m}}_t$ in Eq. (22) is computationally expensive if S is very large. In practice, we use information retrieval (IR) techniques to speed up the computation, which has been suggested by Zhang and Zong [2013]. While standard cross-lingual IR focuses on the relevance between queries and documents and ignores function words (e.g., punctuations), our goal is to find translations in the set of English phrases. Both content and function words are important for indexing and retrieval. Therefore, we use translation probabilities instead of term weights such as tf.idf in retrieval for indexing.

		Chinese	English
Dev	Phrases	2,000	4,000
	Vocabulary	2,994	4,366
Test	Phrases	20,000	40,000
	Vocabulary	10,315	13,233
Dict	Entries	1,000	
	Vocabulary	567	798

Table 1: Statistics of the development set (Dev), test set (Test), and parallel dictionary (Dict).

We build a term-document incidence matrix $\mathcal{M} \in \mathbb{R}^{|V_f| \times S}$ for all foreign words, where V_f is the foreign vocabulary. Each element in the matrix $\mathcal{M}(f, s)$ stores the probability that f connects to the s-th English phrase. This can be done by exploiting the translation probabilities of f given the English words in $\mathbf{e}^{(s)}$. For example, suppose the 621-th English phrase is "*the program has been implemented*" and p("*chengxu*"|"*program*") $= 0.58$, we set $\mathcal{M}$("*chengxu*", 621) $= 0.58$. If another English word is also a translation of "*chengxu*", the matrix retains the highest translation probability. Hence, English phrases can be efficiently retrieved in a monolingual way.

For efficiency, we use a coarse-to-fine approach to computing the Viterbi matching for non-empty English phrases. First, the term-document incidence matrix is used to filter most unlikely source phrases and return a coarse set of candidates. Then, phrase translation probabilities are exactly calculated using Eq. (12).

4.3 Updating Model Parameters

Given the Viterbi matching, the count of the length model $c(J|I; F, E)$ is simply

$$\sum_{t=1}^{T}(1 - \delta(\hat{\mathbf{m}}_t, 0))\delta(J^{(t)}, J)\delta(I^{(\hat{\mathbf{m}}_t)}, I) \tag{24}$$

The count of the translation model $c(f|e; F, E)$ is given by

$$\begin{aligned}&\sum_{t=1}^{T}(1 - \delta(\hat{\mathbf{m}}_t, 0))\frac{p(f|e)}{\sum_{i=0}^{I^{(\hat{\mathbf{m}}_t)}} p(f|\mathbf{e}_i^{(\hat{\mathbf{m}}_t)})} \\ &\qquad \times \sum_{j=1}^{J^{(t)}} \delta(f, \mathbf{f}_j^{(t)}) \sum_{i=0}^{I^{(\hat{\mathbf{m}}_t)}} \delta(e, \mathbf{e}_i^{(\hat{\mathbf{m}}_t)}) \\ &+\sigma(f, e, \mathbf{d})\end{aligned} \tag{25}$$

5 Experiments

5.1 Matching Evaluation

To measure how well our approach identifies parallel phrases from non-parallel corpora F and E, we define the *matching accuracy* as

$$\frac{\sum_{t=1}^{T} \delta(\hat{\mathbf{m}}_t, \mathbf{m}_t^*)}{T} \tag{26}$$

where T is number of all foreign phrases, $\hat{\mathbf{m}}$ is the Viterbi matching predicted by a model, and $\mathbf{m}^*$ is the gold-standard matching.

ϵ	accuracy
e^{-10}	52.30
e^{-30}	52.35
e^{-50}	52.30
e^{-70}	51.80
e^{-90}	51.80

Table 2: Effect of empty translation probability ϵ on matching accuracy on the development set.

iteration	log-likelihood	accuracy
1	-88488	38.50
2	-36238	49.75
3	-28568	51.12
4	-26813	51.70
5	-25752	52.00
6	-25281	52.15
7	-25277	52.20
8	-25264	52.25
9	-25253	52.30
10	-24642	52.35

Table 3: Log-likelihood and matching accuracy on the development set. $\epsilon = e^{-30}$.

As annotating gold-standard matching manually is both time-consuming and labor-intensive, we resort to an automatic approach instead. Given a parallel corpus, we extract a set of parallel phrases $\{\langle \mathbf{f}^{(n)}, \mathbf{e}^{(n)} \rangle\}_{n=1}^{N}$, in which the matching between foreign and English phrases are readily available. Then, the parallel phrase set is corrupted by removing and adding foreign and English phrases randomly. Foreign phrases that have no counterparts on the English side, which we refer to as **noises**, are set to connect to the empty English phrase in the gold-standard matching.

Table 1 shows the statistics of the development set, test set, and parallel dictionary. We used Chinese-English parallel corpus consisting 1,200K pairs of sentences from LDC to extract parallel phrases. From the parallel phrases, we constructed two kinds of monolingual corpora: development set and test set. The development set is used to optimize hyperparameters such as the empty phrase translation probability ϵ. It consists of 2,000 Chinese phrases and 4,000 English phrases (2,000 correspond to Chinese phrases and another 2,000 are noises). The Chinese phrase set has 2,994 distinct words and the English phrase set has 4,366 distinct words. The test set contains 20K Chinese phrases and 40K English phrases. The maximum phrase length is set to 7. The translation probability table learned from the parallel corpus serves as a bilingual lexicon, which has 1,000 entries.

We first use the development set to optimize the hyperparameter ϵ. Table 2 shows the effect of empty translation probability ϵ on matching accuracy. The Vierbi EM algorithm runs for 10 iterations. We find that ϵ does not have a significant impact on accuracy. Therefore, we set ϵ to e^{-30} in the following experiments.

Table 3 shows the log-likelihood and matching accuracy on the development set. We set $\epsilon = e^{-30}$ and the Viterbi EM

$\|\mathbf{d}\|$	accuracy
0	0.05
10	0.55
50	2.45
100	6.70
500	34.20
1,000	52.35

Table 4: Effect of dictionary size on matching accuracy on the development set.

|noises|	accuracy
0	59.30
100	57.65
500	55.30
1,000	54.95
1,500	53.45
2,000	52.35

Table 5: Effect of noises on matching accuracy on the development set.

algorithm ran for 10 iterations. Clearly, the log-likelihood has a high correlation with matching accuracy.

Table 4 shows the effect of dictionary size on matching accuracy. Clearly, using larger seed dictionaries improves the accuracy dramatically. We find that the accuracy reaches 95.28% when the dictionary is enlarged to contain 3,000 entries. This can be seen as an *oracle* accuracy since the dictionary with 3,000 entries covers most words in the development set.

Table 5 shows the effect of noises on matching accuracy. We find that the accuracy decreases with the increase of noises. Without noises, our model is able to achieve an accuracy of 59.30%.

Our final result on the test set is 40.18% as shown in Figure 2. The Viterbi EM algorithm ran for 70 iterations, the dictionary size is 1,000, and the empty translation probability $\epsilon = e^{-30}$.

We analyzed the errors made by our model and found that most wrongly identified phrases are actually very close to the gold-standard phrases. For example, given a Chinese phrase "*ta biaoshi jingji*", our model identifies an English phrase "*he says economy*" while the gold-standard phrase is "*he said economy*". Therefore, although the matching accuracy is relatively low, the identified phrase pairs are still valuable for machine translation, as shown in the following section.

5.2 Translation Evaluation

We follow Zhang and Zong [2013] to evaluate our approach on domain adaptation for machine translation. Given an out-domain parallel corpus L and an in-domain non-parallel corpus U, the task is to maximize the translation performance on unseen in-domain text.

Our experiment runs as follows:

1. Train IBM model 1 on the out-domain parallel corpus L and obtain initial model parameters $\boldsymbol{\theta}$;

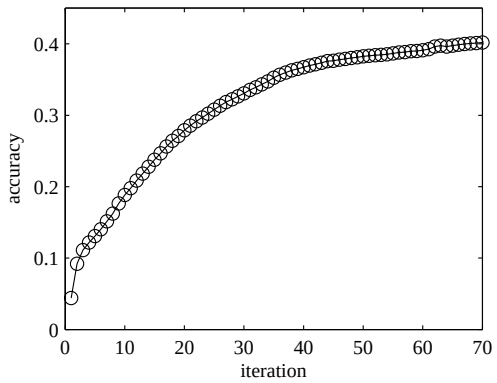

Figure 2: Final matching accuracy on the test set. $|\mathbf{d}| = 1000, \epsilon = e^{-30}$.

2. Identify parallel phrase pairs P from the in-domain non-parallel corpus U using $\boldsymbol{\theta}$;
3. Evaluate the combined parallel corpora $L \cup P$ by training phrase-based models and calculating the BLEU score on unseen in-domain test set;
4. Update model parameters $\boldsymbol{\theta}$ on $L \cup P$;
5. Goto step 2 or terminate if the number of iterations reaches the pre-defined limit (e.g., 5).

The out-domain parallel corpus L (financial articles from FTChina) we used consists of 7,360 pairs of Chinese-English phrases with 41,279 Chinese words and 41,123 English words. The average lengths of Chinese and English phrases are 5.61 and 5.59. The in-domain non-parallel corpus U (news reports from LDC) consists of 2,935,265 Chinese phrases and 3,961,087 English phrases. The average lengths of Chinese and English phrases are 5.67 and 5.91.

We used Moses [Koehn and Hoang, 2007] to learn phrase-based translation models on the combined parallel corpus $L \cup P$ and evaluate translation performance on NIST datasets. We used the SRILM toolkit [Stolcke, 2002] to train a 4-gram language model on the Xinhua portion of the GIGAWORD corpus, which contains 238M English words. The NIST 2002 MT Chinese-English dataset is our the development set and the NIST 2005 dataset is the test set. The evaluation metric is case-insensitive BLEU4 [Papineni *et al.*, 2002].

Table 6 lists the statistics of constructed parallel corpus $L \cup P$ and resulting BLEU scores. At iteration 0, only the out-domain parallel corpus L is used. Then, our system iteratively extracts parallel phrases from the in-domain non-parallel corpus U. We find that the constructed parallel corpus keeps growing by including new phrase pairs that do not exist in L. The size of translation probability table of IBM Model 1 also increases significantly, suggesting that our approach is able to learn translations of unseen words. As a result, our approach achieves statistically significant improvements than using the starting out-domain parallel corpus.

Table 7 shows some example learned parallel phrases. Note that these phrases do not exist in the out-domain parallel corpus L. Words that are unseen in L are highlighted in

iteration	corpus size	Chinese Vocab.	English Vocab.	tTable size	BLEU
0	7,360	4,149	4,134	8,494	7.68
1	43,401	7,794	6,638	31,381	11.21
2	102,396	11,843	9,338	64,748	12.65
3	135,290	13,334	10,474	77,924	12.77
4	148,177	14,059	11,017	82,599	13.23
5	153,681	14,447	11,256	85,279	13.40

Table 6: Results on domain adaptation for machine translation. At iteration 0, we are given a small out-domain parallel corpus with 7,360 phrase pairs. Our system iteratively learns IBM model 1 from the combination of the out-domain parallel corpus and in-domain non-parallel corpus and yields a set of matched in-domain parallel phrases as a byproduct. The quality of constructed parallel corpus is measured by training phrase-based translation models using Moses and evaluating on NIST datasets. Our approach achieves statistically significant improvement than using starting out-domain parallel corpus.

id	Chinese phrase	English phrase
1	*qi zhuyao chanpin shi shiwu*	*its main products are food*
2	*meiguo* ***jiding*** *de zhanlue jihua*	*the set us strategic plan*
3	*quanqiuhua de qushi*	*the trends of* ***globalization***
4	*qianghua* ***lianhe zuozhan*** *gongneng*	*strengthening* ***joint combat*** *functions*
5	***yisilanjiao*** *,* ***tianzhujiao*** *he* ***jidujiao***	***islam*** *,* ***catholicism*** *, and* ***christianity***

Table 7: Example learned parallel phrases. These phrases do not appear in the out-domain parallel corpus L. Words that are unseen in L are highlighted in bold.

bold. We find that 39.27% of learned phrases contain words only seen in L (e.g., No 1 in Table 7), 25.68% contain one word unseen in L (e.g., No 2 and No 3 in Table 7), 19.92% contain two unseen words, and 7.18% contain three unseen words. This finding suggests that our approach is capable of enlarging the vocabulary and learning new phrase pairs.

6 Related Work

Joint Parallel Sentence and Lexicon Extraction. Advocating the "*find-one-get-more*" principle, Fung and Cheung [2004] propose an iterative framework to extract parallel sentences and lexicons from non-parallel corpora via boostrapping and EM. They first use similarity measures to match documents in non-parallel corpora and then extract parallel sentences and new translations from these documents by exploiting bootstrapping on top of IBM model 4. While similar in spirit to their idea, our work assumes that parallel phrases exist in monolingual corpora and develops a new model to capture the correspondence between monolingual phrases in a principled way.

Parallel Sub-Sentential Fragment Extraction. Observing that comparable corpora often contain parallel words or phrases, a number of authors have proposed to extract such parallel fragments from noisy parallel sentences [Munteanu and Marcu, 2006; Quirk *et al.*, 2007; Cettolo *et al.*, 2010]. They first extract candidate parallel sentences from comparable corpora and then identify parallel fragments from the noisy candidate parallel sentences. This approach assumes that comparable corpora are readily available and focuses on finding parallel fragments within comparable sentence pairs, our approach directly builds a generative model on fragments, which can be n-grams extracted from monolingual corpora.

Extracting Parallel Phrases using Lexicons. Our work is also similar to [Zhang and Zong, 2013] that exploits a bilingual lexicon to retrieve parallel phrases from monolingual corpora. The major difference is that their approach is non-iterative. Our approach uses a Viterbi EM algorithm and can improve the learned model iteratively. Experiments show that our iterative approach significantly outperforms the non-iterative approach (see Table 6: 13.40 vs. 11.21).

Translation as Decipherment. Another interesting line of research casts translation with monolingual corpora as a decipherment problem [Ravi and Knight, 2011; Nuhn *et al.*, 2012; Dou and Knight, 2012]. They develop word-based generative models on monolingual corpora and use sampling methods for efficient training. While our approach is also based on word-based translation model, it constructs parallel corpus in the EM loop, which significantly reduces the training complexity.

Transductive Learning on Monolingual Corpora. A number of authors leverage transductive learning to make full use of monolingual data [Ueffing *et al.*, 2007; Bertoldi and Federico, 2009]. They use an existing translation model to translate unseen source-language text. Then, the input and their translations constitute a pseudo parallel corpus. This process iterates until convergence. Zhang and Zong [2013] indicate that this approach can hardly learn new translations consisting of words unseen in the seed parallel corpora.

7 Conclusion

We have presented an iterative approach to learning parallel lexicons and phrases from non-parallel corpora. Experiments show that iterative learning can offer valuable parallel phrases to machine translation systems. In the future, we plan to extend our approach to include more sophisticated alignment models such as IBM models 2-5 and HMM [Vogel *et al.*, 1996]. It is also interesting to scale up to web text and build a never-ending parallel lexicon and phrase mining system.

Acknowledgements

Yang Liu and Maosong Sun are supported by the National Natural Science Foundation of China (No. 61331013 and No. 61432013), the 863 Program (2015AA015407) and Toshiba Corporation Corporate Research & Development Center. Huanbo Luan is supported by the National Natural Science Foundation of China (No. 61303075). This research is also supported by the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme.

References

[Bertoldi and Federico, 2009] Nicola Bertoldi and Marcello Federico. Domain adaptation for statistical machine translation. In *Proceedings of WMT 2009*, 2009.

[Brown *et al.*, 1993] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguisitcs*, 1993.

[Cettolo *et al.*, 2010] Mauro Cettolo, Marcello Federico, and Nicola Bertoldi. Mining parallel fragments from comparable texts. In *Proceedings of IWSLT 2010*, 2010.

[Chiang, 2005] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL 2005*, 2005.

[Daumé III and Jagarlamudi, 2011] Hal Daumé III and Jagadeesh Jagarlamudi. Domain adapation for machine translation by mining unseen words. In *Proceedings of ACL 2011*, 2011.

[Dou and Knight, 2012] Qing Dou and Kevin Knight. Large decipherment for out-of-domain machine translation. In *Proceedings of EMNLP-CONLL 2012*, 2012.

[Dou *et al.*, 2014] Qing Dou, Ashish Vaswani, and Kevin Knight. Beyond parallel data: Joint word alignment and decipherment improves machine translation. In *Proceedings of EMNLP 2014*, 2014.

[Fung and Cheung, 2004] Pascale Fung and Percy Cheung. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In *Proceedings of EMNLP 2004*, 2004.

[Haghighi *et al.*, 2008] Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL 2008*, 2008.

[Klementiev *et al.*, 2012] Alexandre Klementiev, Ann Irvine, Callison-Burch, and David Yarowsky. Toward statistical machine translation without parallel corpora. In *Proceedings of EACL 2012*, 2012.

[Koehn and Hoang, 2007] Philipp Koehn and Hieu Hoang. Factored translation models. In *Proceedings of EMNLP-CoNLL 2007*, 2007.

[Koehn and Knight, 2002] Philipp Koehn and Kevin Knight. Learning a translation lexicon from monolingual corpora. In *Proceedings of ACL 2002 Workshop on Unsupervised Lexical Acquisition*, 2002.

[Koehn *et al.*, 2003] Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, 2003.

[Munteanu and Marcu, 2006] Dragos Stefan Munteanu and Daniel Marcu. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of ACL 2006*, 2006.

[Nuhn *et al.*, 2012] Malte Nuhn, Arne Mauser, and Hermann Ney. Deciphering foreign language by combining language models and context vectors. In *Proceedings of ACL 2012*, 2012.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, 2002.

[Quirk *et al.*, 2007] Chris Quirk, Raghavendra Udupa, and Arul Menzenes. Generative models of noisy translations with applications to parallel fragment extraction. In *Proceedings of MT Summit XI*, 2007.

[Ravi and Knight, 2011] Sujith Ravi and Kevin Knight. Deciphering foreign language. In *Proceedings of ACL 2011*, 2011.

[Stolcke, 2002] Andreas Stolcke. Srilm - an extensible language modeling toolkit. In *Proceedings of ICSLP 2002*, 2002.

[Ueffing *et al.*, 2007] Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. Trasductive learning for statistical machine translation. In *Proceedings of ACL 2007*, 2007.

[Vogel *et al.*, 1996] Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of COLING 1996*, 1996.

[Zhang and Zong, 2013] Jiajun Zhang and Chengqing Zong. Learning a phrase-based translation model from monolingual data with application to domain adaptation. In *Proceedings of ACL 2013*, 2013.