Abstract

Word embeddings are well known to capture linguistic regularities of the language on which they are trained. Researchers also observe that these regularities can transfer across languages. However, previous endeavors to connect separate monolingual word embeddings typically require cross-lingual signals as supervision, either in the form of parallel corpus or seed lexicon. In this work, we show that such cross-lingual connection can actually be established without any form of supervision. We achieve this end by formulating the problem as a natural adversarial game, and investigating techniques that are crucial to successful training. We carry out evaluation on the unsupervised bilingual lexicon induction task. Even though this task appears intrinsically cross-lingual, we are able to demonstrate encouraging performance without any cross-lingual clues.

1 Introduction

As word is the basic unit of a language, the betterment of its representation has significant impact on various natural language processing tasks. Continuous word representations, commonly known as word embeddings, have formed the basis for numerous neural network models since their advent. Their popularity results from the performance boost they bring, which should in turn be attributed to the linguistic regularities they capture (Mikolov et al., 2013b).

Soon following the success on monolingual tasks, the potential of word embeddings for cross-lingual natural language processing has attracted much attention. In their pioneering work, Mikolov et al. (2013a) observe that word embeddings trained separately on monolingual corpora exhibit isomorphic structure across languages, as illustrated in Figure 1. This interesting finding is in line with research on human cognition (Youn et al., 2016). It also means a linear transformation may be established to connect word embedding spaces, allowing word feature transfer. This has far-reaching implication on low-resource scenarios (Daumé III and Jagarlamudi, 2011; Irvine and Callison-Burch, 2013), because word embeddings only require plain text to train, which is the most abundant form of linguistic resource.

However, connecting separate word embedding spaces typically requires supervision from cross-lingual signals. For example, Mikolov et al. (2013a) use five thousand seed word translation pairs to train the linear transformation. In a recent study, Vučić and Korhonen (2016) show that at least hundreds of seed word translation pairs are needed for the model to generalize. This is unfortunate for low-resource languages and domains,
Figure 2: (a) The unidirectional transformation model directly inspired by the adversarial game: The generator $G$ tries to transform source word embeddings (squares) to make them seem like target ones (dots), while the discriminator $D$ tries to classify whether the input embeddings are generated by $G$ or real samples from the target embedding distribution. (b) The bidirectional transformation model. Two generators with tied weights perform transformation between languages. Two separate discriminators are responsible for each language. (c) The adversarial autoencoder model. The generator aims to make the transformed embeddings not only indistinguishable by the discriminator, but also recoverable as measured by the reconstruction loss $L_R$.

because data encoding cross-lingual equivalence is often expensive to obtain.

In this work, we aim to entirely eliminate the need for cross-lingual supervision. Our approach draws inspiration from recent advances in generative adversarial networks (Goodfellow et al., 2014). We first formulate our task in a fashion that naturally admits an adversarial game. Then we propose three models that implement the game, and explore techniques to ensure the success of training. Finally, our evaluation on the bilingual lexicon induction task reveals encouraging performance, even though this task appears formidable without any cross-lingual supervision.

2 Models

In order to induce a bilingual lexicon, we start from two sets of monolingual word embeddings with dimensionality $d$. They are trained separately on two languages. Our goal is to learn a mapping function $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ so that for a source word embedding $x$, $f(x)$ lies close to the embedding of its target language translation $y$. The learned mapping function can then be used to translate each source word $x$ by finding the nearest target embedding to $f(x)$.

We consider $x$ to be drawn from a distribution $p_x$, and similarly $y \sim p_y$. The key intuition here is to find the mapping function to make $f(x)$ seem to follow the distribution $p_y$, for all $x \sim p_x$. From this point of view, we design an adversarial game as illustrated in Figure 2(a): The generator $G$ implements the mapping function $f$, trying to make $f(x)$ passable as target word embeddings, while the discriminator $D$ is a binary classifier striving to distinguish between fake target word embeddings $f(x) \sim p_f(x)$ and real ones $y \sim p_y$. This intuition can be formalized as the minimax game $\min_G \max_D V(D, G)$ with value function

$$ V(D, G) = \mathbb{E}_{y \sim p_y} [\log D(y)] + \mathbb{E}_{x \sim p_x} [\log (1 - D(G(x)))] . \quad (1) $$

Theoretical analysis reveals that adversarial training tries to minimize the Jensen-Shannon divergence $\text{JSD}(p_y || p_f(x))$ (Goodfellow et al., 2014). Importantly, the minimization happens at the distribution level, without requiring word
translation pairs to supervise training.

2.1 Model 1: Unidirectional Transformation

The first model directly implements the adversarial game, as shown in Figure 2(a). As hinted by the isomorphism shown in Figure 1, previous works typically choose the mapping function \( f \) to be a linear map (Mikolov et al., 2013a; Dinu et al., 2015; Lazaridou et al., 2015). We therefore parametrize the generator as a transformation matrix \( G \in \mathbb{R}^{d \times d} \). We also tried non-linear maps parametrized by neural networks, without success. In fact, if the generator is given sufficient capacity, it can in principle learn a constant mapping function to a target word embedding, which makes the discriminator impossible to distinguish, much like the “mode collapse” problem widely observed in the image domain (Radford et al., 2015; Salimans et al., 2016). We therefore believe it is crucial to grant the generator with suitable capacity.

As a generic binary classifier, a standard feedforward neural network with one hidden layer is used to parametrize the discriminator \( D \), and its loss function is the usual cross-entropy loss, as in the value function (1):

\[
L_D = - \log D(y) - \log (1 - D(Gx)) .
\]

(2)

For simplicity, here we write the loss with a mini-batch size of 1; in our experiments we use 128.

The generator loss is given by

\[
L_G = - \log D(Gx) .
\]

(3)

In line with previous work (Goodfellow et al., 2014), we find this loss easier to minimize than the original form \( \log (1 - D(Gx)) \).

Orthogonal Constraint

The above model is very difficult to train. One possible reason is that the parameter search space \( \mathbb{R}^{d \times d} \) for the generator may still be too large. Previous works have attempted to constrain the transformation matrix to be orthogonal (Xing et al., 2015; Zhang et al., 2016b; Artetxe et al., 2016). An orthogonal transformation is also theoretically appealing for its self-consistency (Smith et al., 2017) and numerical stability. However, using constrained optimization for our purpose is cumbersome, so we opt for an orthogonal parametrization (Mhammedi et al., 2016) of the generator instead.

2.2 Model 2: Bidirectional Transformation

The orthogonal parametrization is still quite slow. We can relax the orthogonal constraint and only require the transformation to be self-consistent (Smith et al., 2017): If \( G \) transforms the source word embedding space into the target language space, its transpose \( G^\top \) should transform the target language space back to the source. This can be implemented by two unidirectional models with a tied generator, as illustrated in Figure 2(b). Two separate discriminators are used, with the same cross-entropy loss as Equation (2) used by Model 1. The generator loss is given by

\[
L_G = - \log D_1(Gx) - \log D_2(G^\top y) .
\]

(4)

2.3 Model 3: Adversarial Autoencoder

As another way to relax the orthogonal constraint, we introduce the adversarial autoencoder (Makhzani et al., 2015), depicted in Figure 2(c). After the generator \( G \) transforms a source word embedding \( x \) into a target language representation \( Gx \), we should be able to reconstruct the source word embedding \( x \) by mapping back with \( G^\top \). We therefore introduce the reconstruction loss measured by cosine similarity:

\[
L_R = - \cos \left( x, G^\top Gx \right) .
\]

(5)

Note that this loss will be minimized if \( G \) is orthogonal. With this term included, the loss function for the generator becomes

\[
L_G = - \log D(Gx) - \lambda \cos \left( x, G^\top Gx \right) ,
\]

(6)

where \( \lambda \) is a hyperparameter that balances the two terms. \( \lambda = 0 \) recovers the unidirectional transformation model, while larger \( \lambda \) should enforce a stricter orthogonal constraint.

3 Training Techniques

Generative adversarial networks are notoriously difficult to train, and investigation into stabler training remains a research frontier (Radford et al., 2015; Salimans et al., 2016; Arjovsky and Bottou, 2017). We contribute in this aspect by reporting techniques that are crucial to successful training for our task.

3.1 Regularizing the Discriminator

Recently, it has been suggested to inject noise into the input to the discriminator (Sønderby et al.,
The noise is typically additive Gaussian. Here we explore more possibilities, with the following types of noise, injected into the input and hidden layer:

- Multiplicative Bernoulli noise (dropout) (Srivastava et al., 2014): $\epsilon \sim \text{Bernoulli}(p)$.
- Additive Gaussian noise: $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
- Multiplicative Gaussian noise: $\epsilon \sim \mathcal{N}(1, \sigma^2)$.

As noise injection is a form of regularization (Bishop, 1995; Van der Maaten et al., 2013; Wager et al., 2013), we also try $l_2$ regularization, and directly restricting the hidden layer size to combat overfitting. Our findings include:

- Without regularization, it is not impossible for the optimizer to find a satisfactory parameter configuration, but the hidden layer size has to be tuned carefully. This indicates that a balance of capacity between the generator and discriminator is needed.
- All forms of regularization help training by allowing us to liberally set the hidden layer size to a relatively large value.
- Among the types of regularization, multiplicative Gaussian injected into the input is the most effective, and additive Gaussian is similar. On top of input noise, hidden layer noise helps slightly.

In the following experiments, we inject multiplicative Gaussian into the input and hidden layer of the discriminator with $\sigma = 0.5$.

### 3.2 Model Selection

From a typical training trajectory shown in Figure 3, we observe that training is not convergent. In fact, simply using the model saved at the end of training gives poor performance. Therefore we need a mechanism to select a good model. We observe there are sharp drops of the generator loss $L_G$, and find they correspond to good models, as the discriminator gets confused at these points with its classification accuracy ($D$ accuracy) dropping simultaneously. Interestingly, the reconstruction loss $L_R$ and the value of $\|G^\top G - I\|_F$ exhibit synchronous drops, even if we use the unidirectional transformation model ($\lambda = 0$). This means a good transformation matrix is indeed nearly orthogonal, and justifies our encouragement of $G$ towards orthogonality. With this finding, we can train for sufficient steps and save the model with the lowest generator loss.

As we aim to find the cross-lingual transformation without supervision, it would be ideal to determine hyperparameters without a validation set. The sharp drops can also be indicative in this case. If a hyperparameter configuration is poor, those values will oscillate without a clear drop. Although this criterion is somewhat subjective, we find it to be quite feasible in practice.

### 3.3 Other Training Details

Our approach takes monolingual word embeddings as input. We train the CBOW model (Mikolov et al., 2013b) with default hyperparameters in word2vec.\(^1\) The embedding dimension $d$ is 50 unless stated otherwise. Before feeding them into our system, we normalize the word embeddings to unit length. When sampling words for adversarial training, we penalize frequent words in a way similar to (Mikolov et al., 2013b). $G$ is

\(^1\)https://code.google.com/archive/p/word2vec
initialized with a random orthogonal matrix. The hidden layer size of \( D \) is 500. Adversarial training involves alternate gradient update of the generator and discriminator, which we implement with a simpler variant algorithm described in (Nowozin et al., 2016). Adam (Kingma and Ba, 2014) is used as the optimizer, with default hyperparameters. For the adversarial autoencoder model, \( \lambda = 1 \) generally works well, but \( \lambda = 10 \) appears stabler for the low-resource Turkish-English setting.

4 Experiments

We evaluate the quality of the cross-lingual embedding transformation on the bilingual lexicon induction task. After a source word embedding is transformed into the target space, its \( M \) nearest target embeddings (in terms of cosine similarity) are retrieved, and compared against the entry in a ground truth bilingual lexicon. Performance is measured by top-\( M \) accuracy (Vulić and Moens, 2013): If any of the \( M \) translations is found in the ground truth bilingual lexicon, the source word is considered to be handled correctly, and the accuracy is calculated as the percentage of correctly translated source words. We generally report the harshest top-1 accuracy, unless when comparing with published figures in Section 4.4.

Baselines

Almost all approaches to bilingual lexicon induction from non-parallel data depend on seed lexica. An exception is decipherment (Dou and Knight, 2012; Dou et al., 2015), and we use it as our baseline. The decipherment approach is not based on distributional semantics, but rather views the source language as a cipher for the target language, and attempts to learn a statistical model to decipher the source language. We run the Monogiza system as recommended by the toolkit.\(^3\) It can also utilize monolingual embeddings (Dou et al., 2015); in this case, we use the same embeddings as the input to our approach.

Sharing the underlying spirit with our approach, related methods also build upon monolingual word embeddings and find transformation to link different languages. Although they need seed word translation pairs to train and thus not directly comparable, we report their performance with 50 and 100 seeds for reference. These methods are:

\(^3\)http://www.isi.edu/natural-language/software/monogiza_release_v1.0.tar.gz

<table>
<thead>
<tr>
<th>Wikipedia comparable corpora</th>
<th># tokens</th>
<th>vocab. size</th>
</tr>
</thead>
<tbody>
<tr>
<td>zh-en</td>
<td>zh 21m</td>
<td>3,349</td>
</tr>
<tr>
<td></td>
<td>en 53m</td>
<td>5,154</td>
</tr>
<tr>
<td>es-en</td>
<td>es 61m</td>
<td>4,774</td>
</tr>
<tr>
<td></td>
<td>en 95m</td>
<td>6,637</td>
</tr>
<tr>
<td>it-en</td>
<td>it 73m</td>
<td>8,490</td>
</tr>
<tr>
<td></td>
<td>en 93m</td>
<td>6,597</td>
</tr>
<tr>
<td>ja-zh</td>
<td>ja 38m</td>
<td>6,043</td>
</tr>
<tr>
<td></td>
<td>zh 16m</td>
<td>2,814</td>
</tr>
<tr>
<td>tr-en</td>
<td>tr 6m</td>
<td>7,482</td>
</tr>
<tr>
<td></td>
<td>en 28m</td>
<td>13,220</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Large-scale settings</th>
<th># tokens</th>
<th>vocab. size</th>
</tr>
</thead>
<tbody>
<tr>
<td>zh-en</td>
<td>zh 143m</td>
<td>14,686</td>
</tr>
<tr>
<td></td>
<td>en 1,907m</td>
<td>61,899</td>
</tr>
<tr>
<td>zh-en</td>
<td>zh 2,148m</td>
<td>45,958</td>
</tr>
<tr>
<td></td>
<td>en 5,017m</td>
<td>73,504</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the non-parallel corpora. Language codes: zh = Chinese, en = English, es = Spanish, it = Italian, ja = Japanese, tr = Turkish.

- Translation matrix (TM) (Mikolov et al., 2013a): the pioneer of this type of methods mentioned in the introduction, using linear transformation. We use a publicly available implementation.\(^3\)

- Isometric alignment (IA) (Zhang et al., 2016b): an extension of TM by augmenting its learning objective with the isometric (orthogonal) constraint. Although Zhang et al. (2016b) had subsequent steps for their POS tagging task, it could be used for bilingual lexicon induction as well.

We ensure the same input embeddings for these methods and ours.

The seed word translation pairs are obtained as follows. First, we ask Google Translate\(^4\) to translate the source language vocabulary. Then the target translations are queried again and translated back to the source language, and those that do not match the original source words are discarded. This helps to ensure the translation quality. Finally, the translations are discarded if they fall out of our target language vocabulary.

\(^3\)http://clic.cimec.unitn.it/~georgiana.dinu/download

\(^4\)https://translate.google.com
<table>
<thead>
<tr>
<th>method</th>
<th># seeds</th>
<th>accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MonoGiza w/o emb.</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>MonoGiza w/ emb.</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>TM</td>
<td>50</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>21.79</td>
</tr>
<tr>
<td>IA</td>
<td>50</td>
<td>18.71</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>32.29</td>
</tr>
<tr>
<td>Model 1</td>
<td>0</td>
<td>39.25</td>
</tr>
<tr>
<td>Model 1 + ortho.</td>
<td>0</td>
<td>28.62</td>
</tr>
<tr>
<td>Model 2</td>
<td>0</td>
<td>40.28</td>
</tr>
<tr>
<td>Model 3</td>
<td>0</td>
<td>43.31</td>
</tr>
</tbody>
</table>

Table 2: Chinese-English top-1 accuracies of the MonoGiza baseline and our models, along with the translation matrix (TM) and isometric alignment (IA) methods that utilize 50 and 100 seeds.

4.1 Experiments on Chinese-English

Data

For this set of experiments, the data for training word embeddings comes from Wikipedia comparable corpora. Following (Vulić and Moens, 2013), we retain only nouns with at least 1,000 occurrences. For the Chinese side, we first use OpenCC to normalize characters to be simplified, and then perform Chinese word segmentation and POS tagging with THULAC. The preprocessing of the English side involves tokenization, POS tagging, lemmatization, and lowercasing, which we carry out with the NLTK toolkit. The statistics of the final training data is given in Table 1, along with the other experimental settings.

As the ground truth bilingual lexicon for evaluation, we use Chinese-English Translation Lexicon Version 3.0 (LDC2002L27).

Overall Performance

Table 2 lists the performance of the MonoGiza baseline and our four variants of adversarial training. MonoGiza obtains low performance, likely due to the harsh evaluation protocol (cf. Section 4.4). Providing it with syntactic information can help (Dou and Knight, 2013), but in a low-resource scenario with zero cross-lingual information, parsers are likely to be inaccurate or even unavailable.

The unidirectional transformation model attains reasonable accuracy if trained successfully, but it is rather sensitive to hyperparameters and initialization. This training difficulty motivates our orthogonal constraint. But imposing a strict orthogonal constraint hurts performance. It is also about 20 times slower even though we utilize orthogonal parametrization instead of constrained optimization. The last two models represent different relaxations of the orthogonal constraint, and the adversarial autoencoder model achieves the best performance. We therefore use it in our following experiments. Table 3 lists some word translation examples given by the adversarial autoencoder model.

Comparison With Seed-Based Methods

In this section, we investigate how many seeds TM and IA require to attain the performance level of our approach. There are a total of 1,280 seed translation pairs for Chinese-English, which are removed from the test set during the evaluation for this experiment. We use the most frequent S pairs for TM and IA.

Figure 4 shows the accuracies with respect to...
Table 4: Top-1 accuracies (%) of the MonoGiza baseline and our approach on Spanish-English, Italian-English, Japanese-Chinese, and Turkish-English. The results for translation matrix (TM) and isometric alignment (IA) using 50 and 100 seeds are also listed.

4.2 Experiments on Other Language Pairs

Data
We also induce bilingual lexica from Wikipedia comparable corpora for the following language pairs: Spanish-English, Italian-English, Japanese-Chinese, and Turkish-English. For Spanish-English and Italian-English, we choose to use TreeTagger\(^9\) for preprocessing, as in (Vulić and Moens, 2013). For the Japanese corpus, we use MeCab\(^10\) for word segmentation and POS tagging. For Turkish, we utilize the preprocessing tools (tokenization and POS tagging) provided in LORELEI Language Packs (Strassel and Tracey, 2016), and its English side is preprocessed by NLTK. Unlike the other language pairs, the frequency cutoff threshold for Turkish-English is 100, as the amount of data is relatively small.

The ground truth bilingual lexica for Spanish-English and Italian-English are obtained from Open Multilingual WordNet\(^11\) through NLTK. For Japanese-Chinese, we use an in-house lexicon. For Turkish-English, we build a set of ground truth translation pairs in the same way as how we obtain seed word translation pairs from Google Translate, described above.

Results
As shown in Table 4, the MonoGiza baseline still does not work well on these language pairs, while our approach achieves much better performance. The accuracies are particularly high for Spanish-English and Italian-English, likely because they are closely related languages, and their embedding spaces may exhibit stronger isomorphism. The

Effect of Embedding Dimension
As our approach takes monolingual word embeddings as input, it is conceivable that their quality significantly affects how well the two spaces can be connected by a linear map. We look into this aspect by varying the embedding dimension \(d\) in Figure 5. As the dimension increases, the accuracy improves and gradually levels off. This indicates that too low a dimension hampers the encoding of linguistic information drawn from the corpus, and it is advisable to use a sufficiently large dimension.

\(^9\)http://www.cis.uni-muenchen.de/~schmid/tools/
\(^10\)http://taku910.github.io/mecab
\(^11\)http://compling.hss.ntu.edu.sg/omnw
Table 5: Top-1 accuracies (%) of our approach to inducing bilingual lexica for Chinese-English from Wikipedia and Gigaword. Also listed are results for translation matrix (TM) and isometric alignment (IA) using 50 and 100 seeds.

<table>
<thead>
<tr>
<th>method</th>
<th># seeds</th>
<th>Wikipedia</th>
<th>Gigaword</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM</td>
<td>50</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>4.79</td>
<td>2.07</td>
</tr>
<tr>
<td>IA</td>
<td>50</td>
<td>3.25</td>
<td>1.68</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>7.08</td>
<td>4.18</td>
</tr>
<tr>
<td>Ours</td>
<td>0</td>
<td>7.92</td>
<td>2.53</td>
</tr>
</tbody>
</table>

Table 6: Top-5 accuracies (%) of 5k and 10k most frequent words in the French-English setting. The figures for the baselines are taken from (Cao et al., 2016).

<table>
<thead>
<tr>
<th>method</th>
<th>5k</th>
<th>10k</th>
</tr>
</thead>
<tbody>
<tr>
<td>MonoGiza w/o embeddings</td>
<td>13.74</td>
<td>7.80</td>
</tr>
<tr>
<td>MonoGiza w/ embeddings</td>
<td>17.98</td>
<td>10.56</td>
</tr>
<tr>
<td>(Cao et al., 2016)</td>
<td>23.54</td>
<td>17.82</td>
</tr>
<tr>
<td>Ours</td>
<td>68.59</td>
<td>51.86</td>
</tr>
</tbody>
</table>

configuration used in our experiments, the adversarial autoencoder model takes about two hours to train for 500k minibatches on a single CPU.

4.4 Comparison With (Cao et al., 2016)

In order to compare with the recent method by Cao et al. (2016), which also uses zero cross-lingual signal to connect monolingual embeddings, we replicate their French-English experiment to test our approach. This experimental setting has important differences from the above ones, mostly in the evaluation protocol. Apart from using top-5 accuracy as the evaluation metric, the ground truth bilingual lexicon is obtained by performing word alignment on a parallel corpus. We find this automatically constructed bilingual lexicon to be noisier than the ones we use for the other language pairs; it often lists tens of translations for a source word. This lenient evaluation protocol should explain MonoGiza’s higher numbers in Table 6 than what we report in the other experiments. In this setting, our approach is able to considerably outperform both MonoGiza and the method by Cao et al. (2016).

5 Related Work

5.1 Cross-Lingual Word Embeddings for Bilingual Lexicon Induction

Inducing bilingual lexica from non-parallel data is a long-standing cross-lingual task. Except for the decipherment approach, traditional statistical methods all require cross-lingual signals (Rapp, 1999; Koehn and Knight, 2002; Fung and Cheung, 2004; Gaussier et al., 2004; Haghighi et al., 2008; Vulić et al., 2011; Vulić and Moens, 2013).

Recent advances in cross-lingual word embeddings (Vulić and Korhonen, 2016; Upadhyay et al., 2016)
have rekindled interest in bilingual lexicon induction. Like their traditional counterparts, these embedding-based methods require cross-lingual signals encoded in parallel data, aligned at document level (Vulić and Moens, 2015), sentence level (Zou et al., 2013; Chandar A P et al., 2014; Hermann and Blunsom, 2014; Kočiský et al., 2014; Gouws et al., 2015; Luong et al., 2015; Coulmance et al., 2015; Oshikiri et al., 2016), or word level (i.e. seed lexicon) (Gouws and Søgaard, 2015; Wick et al., 2016; Duong et al., 2016; Shi et al., 2015; Mikolov et al., 2013a; Dinu et al., 2015; Lazaridou et al., 2015; Faruqui and Dyer, 2014; Lu et al., 2015; Ammar et al., 2016; Zhang et al., 2016a, 2017; Smith et al., 2017). In contrast, our work completely removes the need for cross-lingual signals to connect monolingual word embeddings, trained on non-parallel text corpora.

As one of our baselines, the method by Cao et al. (2016) also does not require cross-lingual signals to train bilingual word embeddings. It modifies the objective for training embeddings, whereas our approach uses monolingual embeddings trained beforehand and held fixed. More importantly, its learning mechanism is substantially different from ours. It encourages word embeddings from different languages to lie in the shared semantic space by matching the mean and variance of the hidden states, assumed to follow a Gaussian distribution, which is hard to justify. Our approach does not make any assumptions and directly matches the mapped source embedding distribution with the target distribution by adversarial training.

A recent work also attempts adversarial training for cross-lingual embedding transformation (Barone, 2016). The model architectures are similar to ours, but the reported results are not positive. We tried the publicly available code on our data, but the results were not positive, either. Therefore, we attribute the outcome to the difference in the loss and training techniques, but not the model architectures or data.

5.2 Adversarial Training

Generative adversarial networks are originally proposed for generating realistic images as an implicit generative model, but the adversarial training technique for matching distributions is generalizable to much more tasks, including natural language processing. For example, Ganin et al. (2016) address domain adaptation by adversarially training features to be domain invariant, and test on sentiment classification. Chen et al. (2016) extend this idea to cross-lingual sentiment classification. Our research deals with unsupervised bilingual lexicon induction based on word embeddings, and therefore works with word embedding distributions, which are more interpretable than the neural feature space of classifiers in the above works.

In the field of neural machine translation, a recent work (He et al., 2016) proposes dual learning, which also involves a two-agent game and therefore bears conceptual resemblance to the adversarial training idea. The framework is carried out with reinforcement learning, and thus differs greatly in implementation from adversarial training.

6 Conclusion

In this work, we demonstrate the feasibility of connecting word embeddings of different languages without any cross-lingual signal. This is achieved by matching the distributions of the transformed source language embeddings and target ones via adversarial training. The success of our approach signifies the existence of universal lexical semantic structure across languages. Our work also opens up opportunities for the processing of extremely low-resource languages and domains that lack parallel data completely.

Our work is likely to benefit from advances in techniques that further stabilize adversarial training. Future work also includes investigating other divergences that adversarial training can minimize (Nowozin et al., 2016), and broader mathematical tools that match distributions (Mohamed and Lakshminarayanan, 2016).

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China (No. 61522204), the 973 Program (2014CB340501), and the National Natural Science Foundation of China (No. 61331013). 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


