



Learning Bilingual Word Embedding Mappings with Similar Words in Related Languages Using GAN

Ghafour Alipour, Jamshid Bagherzadeh Mohasefi & Mohammad-Reza Feizi-Derakhshi

To cite this article: Ghafour Alipour, Jamshid Bagherzadeh Mohasefi & Mohammad-Reza Feizi-Derakhshi (2022) Learning Bilingual Word Embedding Mappings with Similar Words in Related Languages Using GAN, Applied Artificial Intelligence, 36:1, 2019885, DOI: 10.1080/08839514.2021.2019885

To link to this article: <https://doi.org/10.1080/08839514.2021.2019885>



© 2022 The Author(s). Published with license by Taylor & Francis Group, LLC.



Published online: 08 Feb 2022.



Submit your article to this journal [↗](#)



Article views: 1278



View related articles [↗](#)




View Crossmark data [↗](#)



Citing articles: 1 View citing articles [↗](#)

Learning Bilingual Word Embedding Mappings with Similar Words in Related Languages Using GAN

Ghafour Alipour ^a, Jamshid Bagherzadeh Mohasefi^b, and Mohammad-Reza Feizi-Derakhshi^{a,c}

^aDepartment of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran; ^bDepartment of Computer Engineering, Urmia University, Urmia, Iran; ^cFaculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

ABSTRACT

Cross-lingual word embeddings display words from different languages in the same vector space. They provide reasoning about semantics, compare the meaning of words across languages and word meaning in multilingual contexts, necessary to bilingual lexicon induction, machine translation, and cross-lingual information retrieval. This paper proposes an efficient approach to learn bilingual transform mapping between monolingual word embeddings in language pairs. We choose ten different languages from three different language families and downloaded their last update Wikipedia dumps¹ Then, with some pre-processing steps and using word2vec, we produce word embeddings for them. We select seven language pairs from chosen languages. Since the selected languages are relative, they have thousands of identical words with similar meanings. With these identical dictation words and word embedding models of each language, we create training, validation and test sets for the language pairs. We then use a generative adversarial network (GAN) to learn the transform mapping between word embeddings of source and target languages. The average accuracy of our proposed method in all language pairs is 71.34%. The highest accuracy is achieved for the Turkish-Azerbaijani language pair with the accuracy 78.32%, which is noticeably higher than prior methods.

ARTICLE HISTORY

Received 14 January 2021
Revised 23 November 2021
Accepted 9 December 2021

Introduction

Nowadays, there are several ways to represent language words. One broadly used word representation method is word embeddings, which connects the human understanding of language to a machine and is crucial to solving many natural-language-processing (NLP) problems. Word embedding is a common method to learn word representation where words with close meaning have close representations (Mikolov et al. 2013b; Pennington, Socher, and Manning 2014). Some traditional methods, such as one-hot encoding and bag of words, are helping some machine learning (ML) tasks, but they are un-ordered, and

CONTACT Jamshid Bagherzadeh Mohasefi  j.bagherzadeh@urmia.ac.ir  Department of Computer Engineering, Urmia University, Urmia, Iran

© 2022 The Author(s). Published with license by Taylor & Francis Group, LLC.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

therefore, the context and frequency of words are lost. Nevertheless, these methods do not give any information about the meaning (semantics) and the structural relationships between words (syntax) (Duong et al. 2016). In word embedding, all the words in a language are represented in n -dimensional space with real-valued numbers, where each number draws a dimension of the word's meaning. As a result, semantically close words have close vectors and vice versa.

A method, proposed by Google researchers, for learning word embeddings is based on either the skip-gram or the continuous bag-of-words (CBOW) architectures, which are implemented in Word2vec (Mikolov et al. 2013b) and Fast-Text (Joulin et al. 2016) libraries. FastText is an extension of Word2vec, representing sentences with a bag of words, a bag of n -grams, sub-word information, and sharing information across classes through a hidden representation. Another approach, proposed by Stanford university researchers, is Glove, which is achieved by mapping words into a latent space where the distance between words is related to semantic similarity (Pennington, Socher, and Manning 2014).

In many NLP tasks, especially in Neural Machine Translation (NMT) (Bahdanau, Cho, and Bengio 2016), monolingual word vectors are trained independently for each language on its corpora. And then, these monolingual vectors map to a shared space on a bilingual dictionary (Lazaridou, Dinu, and Baroni 2015; Mikolov et al. 2013a). There is a structural similarity between word embedding spaces across the source and target languages, so their mapping is worthwhile. The mapping between word vectors is known as cross-lingual word embedding model, which enables cross-lingual information transfer. Cross-lingual word embedding is a natural extension facilitates several cross-lingual applications, such as sentiment analysis, dependency parsing, and machine translation.

There is an excellent demand for cross-lingual word embedding models in the broad majority of language pairs, including a resource-lean language (e.g., Turkmen) with a resource-rich language (e.g., Turkish, France). Furthermore, there are no cross-lingual word embedding models for many combinations of significant resource-rich languages (e.g., Spanish-Russian). Recently, some methods have been suggested for cross-lingual word embedding models. In most of these methods, large parallel corpora or sizable dictionaries with high-quality bilingual word embedding models have been used to learn a high-performance mapping between languages (Ammar et al. 2016; Gouws, Bengio, and Corrado 2015; Vulić and Moens 2015).

A critical obstacle toward bilingual transfer is lexical matching between the source and the target languages. Such lexical matchings are not prepared for most languages and dialect pairs, so discovering word mappings with no prior knowledge is extremely valuable for cross-lingual

applications. Prior works have focused on independently trained word embeddings in each language by monolingual corpora. They learn a linear transformation to map the embeddings using a small or medium-sized lexical matching as a bilingual seed dictionary from the source language to the target language (Artetxe, Labaka, and Agirre 2016). The ability to produce lexical items of two different languages in a shared cross-lingual space leads the NLP research further. Word-level connections between languages are used in transferred statistical parsing (Ammar et al. 2016; Zeman et al. 2018) or language understanding systems (Mrkšić et al. 2017) and later by using a tiny seed bilingual dictionary (Artetxe, Labaka, and Agirre 2016; Kondrak, Hauer, and Nicolai 2017). However, they do not satisfactorily handle good accuracy and need more labeled data to get better results.

To succeed in lexical matching in language pairs' problem, we perform the exact dictated words in our experiments that increase accuracy without labeled data. Notably, if the source and the target languages are relevant or come from a common language family, they have much mutual intelligibility. By applying some pre-processing steps, we increase lexical matching among pair languages. Since the number of the exact dictated words is significant, we use them to learn a neural network to find a nonlinear mapping between word vectors of languages.

This paper presents a new approach to studying bilingual word embeddings mapping between related languages. First, we use Wikipedia XML dumps for each language as the text source and extract tokens in each language. Next, we use the Word2vec model library to produce word embeddings. Then, we obtain the words with the same dictation between language pairs. Finally, we train our model using the results obtained in the previous step, to find the mapping between word embeddings. The contributions of this paper are:

- To improve the bilingual word embedding mapping method between languages.
- To find nonlinear transformation mappings, especially for low-resource and relative languages.
- The proposed model is based on recent research on the combination of neural machine translation encoder-decoder and GAN models.
- Our proposed model augments a 4-layer BLSTM encoder-decoder with an attention mechanism, taking context into the model to learn bilingual word mappings and complete bilingual word embeddings.
- A convolutional neural network implements our proposed model discriminator to distinguish real target vectors.
- We design a list of experiments on seven language pairs. Our experimental results demonstrate a significant advantages of learning word mapping in related languages.

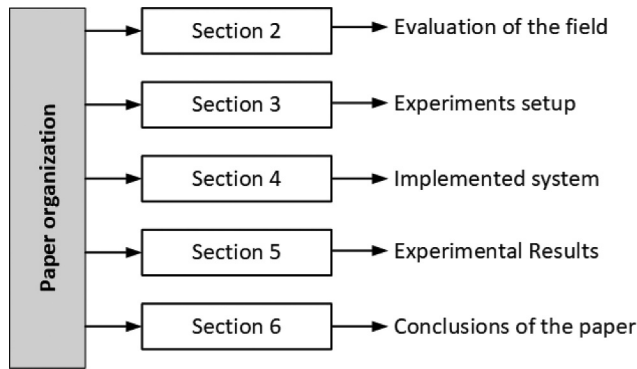


Figure 1. Overall organization of the paper.

The structure of our paper is illustrated in [Figure 1](#). The rest of the paper proceeds as follows. First, we present some essential points and the evolution of the cross-lingual word embedding models in [Section 2](#). Next, in [Section 3](#), the method for data collection and experimental setup are detailed. [Section 4](#), describes the implemented system. Next, in [Section 5](#), our experimental results are illustrated. Finally, we conclude our paper results in [Section 6](#).

The Evaluation of the Field

Most cross-lingual word embedding models are created and extended using monolingual word embedding models (Vulić and Moens 2015). At first, the model learns word embedding vectors for each language words using its large monolingual corpora. Then, it retains a mapping from the source language word embeddings to the target language word embeddings. In the next section, we briefly review the monolingual word embedding models.

Word Embedding Models

Word2vec (Mikolov et al. 2013b) is a shallow neural network with two layers to produce word embeddings in a language. It receives a massive corpus of text documents as input and creates a vector space where each word in the corpus keeps in touch with a vector in the space. Word vectors in the vector space have a specification that semantically close words in the corpus have close vectors in the space. Word2vec is implemented in two structures: Skip-gram and continuous bag-of-words (CBOW).

Skip-gram with negative sampling (Mikolov et al. 2013b) is a popular model due to its robustness and training performance (Levy, Goldberg, and Dagan 2015). It produces a language model by converging on learning effective representation instead of modeling word probabilities accurately. It provides

word vectors that are good at predicting the surrounding context words by offering a source word. The model minimizes the following skip-gram objective, using training data:

$$L_{SG} = -\frac{1}{N} \sum_{t=1}^N \sum_{-C \leq j \leq C, j \neq 0} \log P(w_{t+j} | w_t) \quad (1)$$

N is the number of words in the training corpus, and C is the context window's size. The reverse of Skip-gram is Continuous Bag of Words (CBOW). It tries to produce a source word according to the surrounding words. CBOW minimizes the following objective in training data.

$$L_{CBOW} = -\frac{1}{N} \sum_{t=1}^N \log P(w_t | w_{t-C}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+C}) \quad (2)$$

The skip-gram architecture weighs nearby context words more heavily than more distant context words. According to the authors (Mikolov et al. 2013a), CBOW is faster while skip-gram is slower but better for infrequent words. Explained models are shown in Figure 2.

The Global Vectors for Word Representation (GloVe) (Pennington, Socher, and Manning 2014) extends the Word2vec method; and efficiently learns word vectors. Word2vec and GloVe do the same things and perform similarly in NLP tasks. The notable difference is the way they are built. Word2vec builds word embeddings using a predictive model, while GloVe is a count-based model. Glove learns to make a co-occurrence matrix by counting the frequency of appearing a word in a context.

FastText (Bojanowsk et al. 2017), another extension of the Word2vec model, handles each word as a composition of character n-grams and not tokens. For example, with different representations of “school” and “house,” we can build a representation for “schoolhouse,” which would otherwise appear too infrequently to learn dictionary-level embeddings. This difference enables FastText to generate better word embeddings for rare words and out of vocabulary words. Both Glove and Word2vec cannot generate highly efficient word embeddings for rare words.

Cross-Lingual Mapping-based Approaches

Mapping-based approaches try to learn a mapping between monolingual representations of two languages. In these approaches, first, the method trains monolingual word embeddings on massive monolingual corpora. Then they learn a transformation matrix between monolingual representations in different languages to map unknown words of languages. They frequently generate a list of word pairs between the source and the target

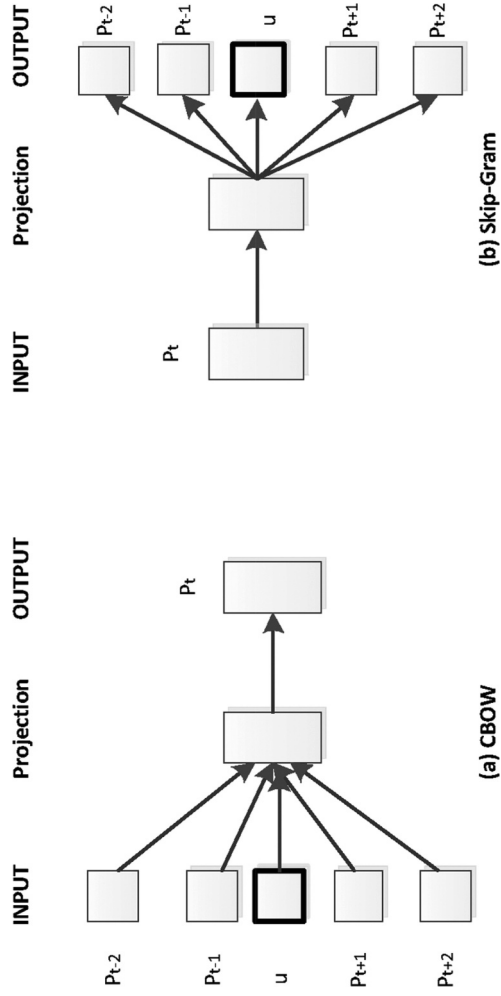


Figure 2. The architecture of CBOW and Skip-gram as described in (Mikolov et al. 2013b).

languages that they translate. There are four types of mapping-based word embedding approaches proposed (Ruder, Vulic, and Søgaard 2019): Regression methods, Orthogonal methods, Canonical methods, and Margin methods.

Regression methods are the most powerful methods for learning a linear transformation between word embeddings of source and target languages by maximizing their similarity. Mikolov, Le, and Sutskever (2013) noted that the words and their translations have similar geometric relations in monolingual word embeddings if a suitable linear transformation is applied.

Orthogonal methods apply orthogonality constraints on the transformation mapping matrix, which improves regression methods' performance. Based on the assumption, the transformation matrix W is orthogonal ($W^T W = I$). The solution is obtained from the singular value decomposition of YX^T .

$$W^* = \arg \min_W \|WX - Y\| = UV^T s.t. U\Sigma V^T = SVD(YX^T) \quad (3)$$

Canonical methods map both languages' word embeddings to a new shared space using Canonical Correlation Analysis (CCA) that maximizes their similarity. Faruqui and Dyer (2014) use CCA to map words from two languages into a shared embedding space.

Margin methods map the source language's word embeddings to maximize the margin between correct translations and other candidates. Lazaridou, Dinu, and Baroni (2015) propose another objective for the linear transformation. They realize that using least-squares as an objective for learning a projection matrix leads to hubness. To find the correct translation vector y_i of a source word x_i , they use a margin-based (max-margin) ranking loss (Collobert and Weston 2008) to train the model. Jinsong Su et al. use graph-based semantic information to learn bilingual word embedding (Jinsong, et al. 2018a).

Creating robust cross-lingual word representations with some parallel data (seed lexicon) is an essential avenue of research. All references in Table 1, have worked on linear transformation.

Unfortunately, most linear transformation mapping approaches are not accurate enough. Therefore, the approaches require a long way to be more precise and reliable. Besides, there are rare efforts in nonlinear transformation mappings. Both Mikolov et al. (2013b) and Conneau et al. (2018) found

Table 1. Word embedding mapping methods.

Regression Methods	Orthogonal methods	Canonical methods	Margin methods
Mikolov et al. 2013a	Xing et al. 2015	Faruqui and Dyer 2014	Lazaridou, Dinu, and Baroni 2015
Hauer et al. 2017	Zhang et al. 2017	Lu et al. 2015	Dinu et al. 2015
Mogadala et al. 2016	Artetxe, Labaka, and Agirre 2018	Ammar et al. 2016	Mrkšić et al. 2017
Conneau et al. 2018	Smith et al.	Rajendran et al. 2016	Calixto et al. 2017

that a linear transformation performs better than a nonlinear transformation learned via a feedforward neural network. Makhzani et al. (2016) use adversarial autoencoders to map word embeddings between languages. The reported performances are weak in comparison to other methods.

Jinsong et al. (2018a) to model the bilingual semantics produce a neural generative autoencoder. Zhang et al. (2020) for cross-lingual embedding mappings use Wasserstein GAN (Arjovsky, Chintala, and Bottou 2017), which combines back-translation with target-side and preliminary mappings learning. Their used dataset was not big enough, and the model requires more iterations to converge on the discriminator as it will be slower to be trained on it.

In brief, there has not been any neural network-based model yet that proves to construct a more effective mapping model on feedforward neural networks. Early cross-lingual word embedding models relied on a large amount of parallel data (Artetxe, Labaka, and Agirre 2016; Mikolov et al. 2013b). Still, more recent methods have tried to minimize the amount of supervision necessary (Artetxe, Labaka, and Agirre 2017; Levy, Søgaard, and Goldberg 2017; Smith et al. ; Vulic´ and Korhonen 2016). Some researchers have presented almost unsupervised methods that do not use any form of cross-lingual supervision data (Conneau et al. 2018; Shigeto et al. 2015; Valerio and Barone 2016; Zhang et al. 2017). Unsupervised cross-lingual word embeddings try to evolve bilingual lexicons and machine translation models without parallel corpora and translations (Duong et al. 2016; Lample et al. 2018).

Recently, approaches have been proposed that learn an initial seed lexicon in a completely unsupervised way. All unsupervised cross-lingual word embeddings methods are based on the mapping approaches. Conneau et al. (2018) learn an initial mapping in an adversarial way by training a discriminator to differentiate between projected and actual target language embeddings. Artetxe et al. (Artetxe, Labaka, and Agirre 2018) propose to use an initialization method based on the heuristic that translations have similar similarity distributions across languages. Hoshen and Wolf (2018) introduced a method with the first project vectors of the N most frequent words to a lower-dimensional space with PCA. Their approach minimizes the sum of Euclidean distances by learning $W^{s \rightarrow t}$ and $W^{t \rightarrow s}$ enforce cyclical consistency constraints that force vectors round-projected to the other language space and back to remain unchanged.

Data Collection and Experimental Setup

Turkic languages are spoken across a wide area, stretching from the Balkans in Europe through Central Asia to northeast Siberia (Hammarström, Forkel, and Haspelmath 2017). There exist several alphabets used by Turkic languages. The Latin alphabet is a well-established alphabet in the Turkic languages today. It is currently used by (with different versions) Turkey, Uzbekistan, Azerbaijan, and Turkmenistan and will be used by Kazakhstan.

Turkish, the official language of Turkey, is the most widely spoken of the Turkic languages and has the biggest articles set in the family inside the Wikipedia dumps. We use Turkish as the source language in our bilingual mapping experiments and Azerbaijani, Turkmen, and Uzbek as the target languages.

The Indo-European languages are among the most major language families and are mostly used in western and southern Eurasia. For our experiments, from the North Germanic branch of the family, we chose Swedish as the source language, Danish and Norwegian languages as the target languages, and from the south Slavic branch, we selected Serbian as a source, Croatian, and Bosnian as the target languages.

One of the first things required for NLP tasks is a corpus that refers to a collection of texts. One of the best rich sources of a well-organized vast amount of non-adversarial textual data is Wikipedia. It is freely and conveniently available online, which makes it a valuable resource to build NLP systems.

By each language Wikipedia text dumps (XML files), we prepared a monolingual corpus for all mentioned languages. For each language, its Wikipedia dump contains just the latest versions of the Wikipedia articles (November 2021). [Table 2](#) shows the number of articles and tokens of the languages.

To construct a text corpus from Wikipedia without article markups, punctuations, and links, we use the WikiCorpus tool from gensim,¹ an XML parser library for Python, which converts Wikipedia dump files to text corpus. To pre-process the text corpus for the Word2Vec model, we convert all the corpus text to lowercase form and delete all the special characters, digits, and extra spaces from the text. After that, we use the Word2vec implementation of the gensim library to provide a monolingual embedding model in each language. As for Word2vec parameters, no lemmatization was done, the window size was set to 5, and the output dimensions were set to 768. We only estimated representation vectors for words, which occurred five times or more in the monolingual corpus. [Figure 3](#) shows the learning process for word vectors in each language.

Table 2. An approximate count of articles and tokens in Wikipedia dumps for each language (K = 1000).

Language	Number of articles	Number of tokens
Turkish	448k	492 K
Azerbaijani	180 K	282 K
Uzbek	140 K	152 K
Turkmen	23 K	35 K
Swedish	2887 K	917 K
Danish	270k	342 K
Norwegian (Bokmål)	569 K	563k
Serbian	651 K	643 K
Bosnian	88 K	207 K
Croatian	209 K	388 K

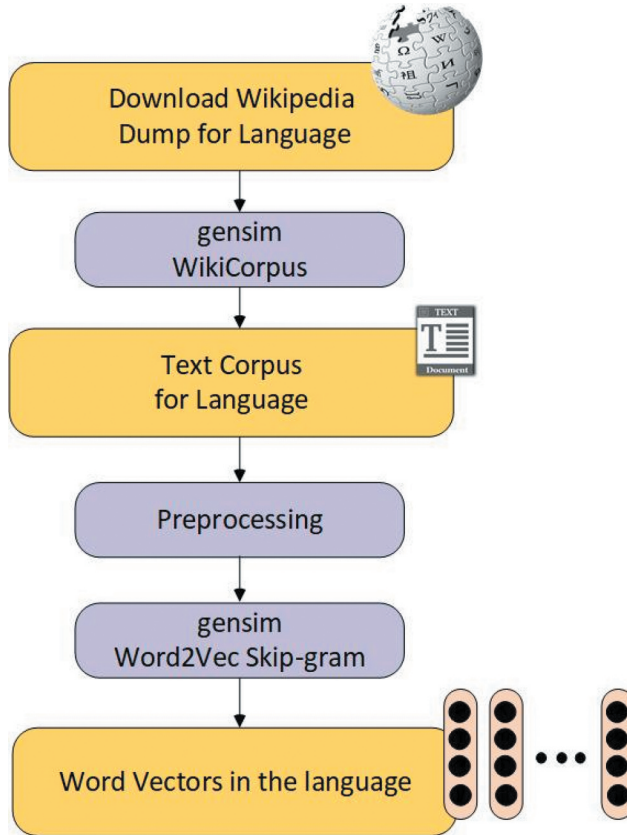


Figure 3. The process of learning word vectors in each language.

In our experiments, there are seven language pairs, Turkish-Azerbaijani, Turkish-Uzbek, Turkish-Turkmen, Swedish-Danish, Swedish-Norwegian, Serbian-Bosnian, and Serbian-Croatian. All language pairs are relevant and use the Latin alphabet; so many words have the same dictation and meaning.

We need a few thousand word pairs as a seed dictionary for better and accurate bilingual word embeddings transformation. Preparing a seed dictionary between languages is usually not easy and requires a lot of cost and effort. On the other hand, a reasonable size seed dictionary makes the final word embedding mapping model more accurate.

We propose choosing the exact dictation words as the bilingual seed dictionary. The underlying assumption is that word embeddings across relative languages share similar local and global arrangements. For example, the distance between the words *Kedi* and *Köpek* in Turkish should be relatively similar to the distance between *Pişik* and *İt* in Azerbaijani. We try to recover the transformation between language pairs using seed dictionaries. We split

Table 3. The number of words in seed dictionaries and size of the training, validation, and test sets ($K = 1000$).

Language Pairs	Seed dictionary Size	Training set size	Validation set size	Test set size
Turkish-Azerbaijani	82 K	52 K	15 K	15k
Turkish-Uzbek	24 K	16 K	4 K	4 K
Turkish-Turkmen	5.5 K	3.5 K	1 K	1 K
Swedish-Danish	167 K	117 K	25 K	25 K
Swedish-Norwegian	254 K	174 K	40 K	40 K
Serbian-Bosnian	111 K	71 K	20 K	20 K
Serbian-Croatian	135 K	91 K	22 K	22 K

each seed dictionary into three parts: a training set, a test set, and a validation set. [Table 3](#) shows the number of the same dictation tokens in the language pairs and the amount of their training set, test set, and validation set.

Implemented System

In this section, we present our proposed network. A brief overview of the proposed network is illustrated in [Figure 4](#). The network includes two main parts. These parts are: A generator network that transfers a word vector from a source language to a target language, and the discriminator network that distinguishes the real/fake word vector.

A GAN (Goodfellow et al. 2014) comprises a generator model, G , and a discriminator model, D . The generator objects to learn a mapping function from a prior noise distribution p_y to an unknown data distribution p_x in the real data space. The discriminator tries to discern between generated and real data. Both networks are trained competing against each other in a min-max game with value function $V(G, D)$:

$$\min_G \max_D V(G, D) = E_{X \rightarrow p_x} [\log(D(X))] + E_{Y \rightarrow p_y} [\log(1 - D(G(Y)))] \quad (4)$$

During training, the generator learns to generate more realistic vectors to deceive the discriminator while the discriminator improves itself to discern the real vectors from the generated one. Our GAN model is mainly focused on learning one-to-one mappings from an input vector to an output vector.

Let $\mathbf{X} = \{x_1, x_2, \dots, x_{|X|}\}$ be the vocabulary of a source language S_i with $|X|$ words, and $\mathbf{X} \in \mathbb{R}^{|X| \times l}$ be the corresponding word embeddings of length l and let $\mathbf{Y} = \{y_1, y_2, \dots, y_{|Y|}\}$ be the vocabulary of the target language T_j with $|Y|$ words, and $\mathbf{Y} \in \mathbb{R}^{|Y| \times m}$ is the corresponding word embeddings of length m . We denote the word vector for a word x by \mathbf{X} .

The source and target languages are aligned with a seed lexicon dictionary (binary matrix) D so that $D_{ij} = 1$ if the i -th word in the source language is aligned with the j -th word in the target language. Our objective is to find the

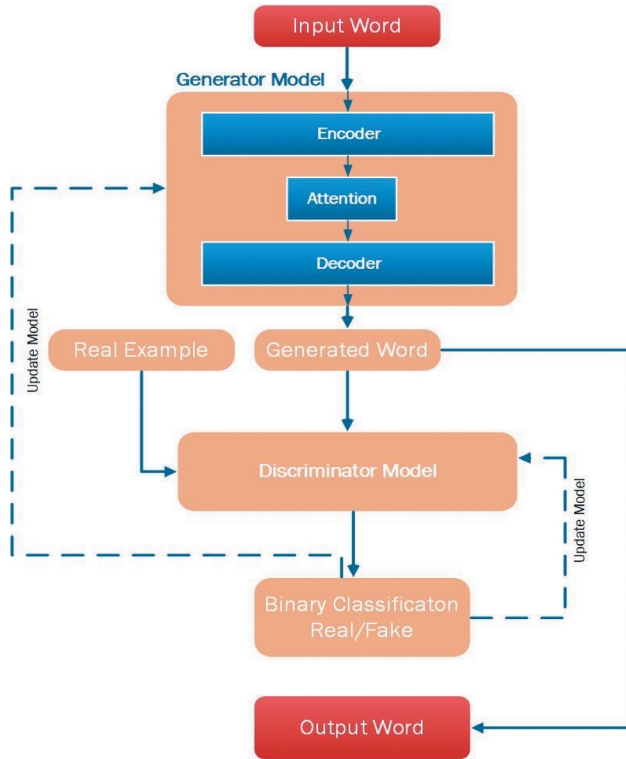


Figure 4. Overview of the proposed model.

dictionary matrix D by learning the mapping matrix W , which transforms input language word embeddings X to the target language word embeddings Y . Our bilingual word embeddings training algorithm is as follows:

The generator consists of an encoder-decoder architecture with an attention mechanism (Bahdanau, Cho, and Bengio 2016; Luong and Manning 2016), as shown in Figure 5.

In our experiments, encoder and decoder networks are recurrent neural networks (RNN) implemented by stacking multiple Bi-directional Long Short-Term Memory (BLSTM) layers. The encoder reads the source word embedding vector \mathbf{x}_k and produces a high-level representation $\mathbf{h} = \{h_1, h_2, \dots, h_m\}$:

$$\mathbf{h} = \text{Encoder}(\mathbf{X}) \quad (5)$$

The decoder network reads the encoding and generates an output sequence in the target language word embeddings space. Attention is a mechanism that gives a richer encoding of the source sequence to construct a context vector used by the decoder. The decoder calculates

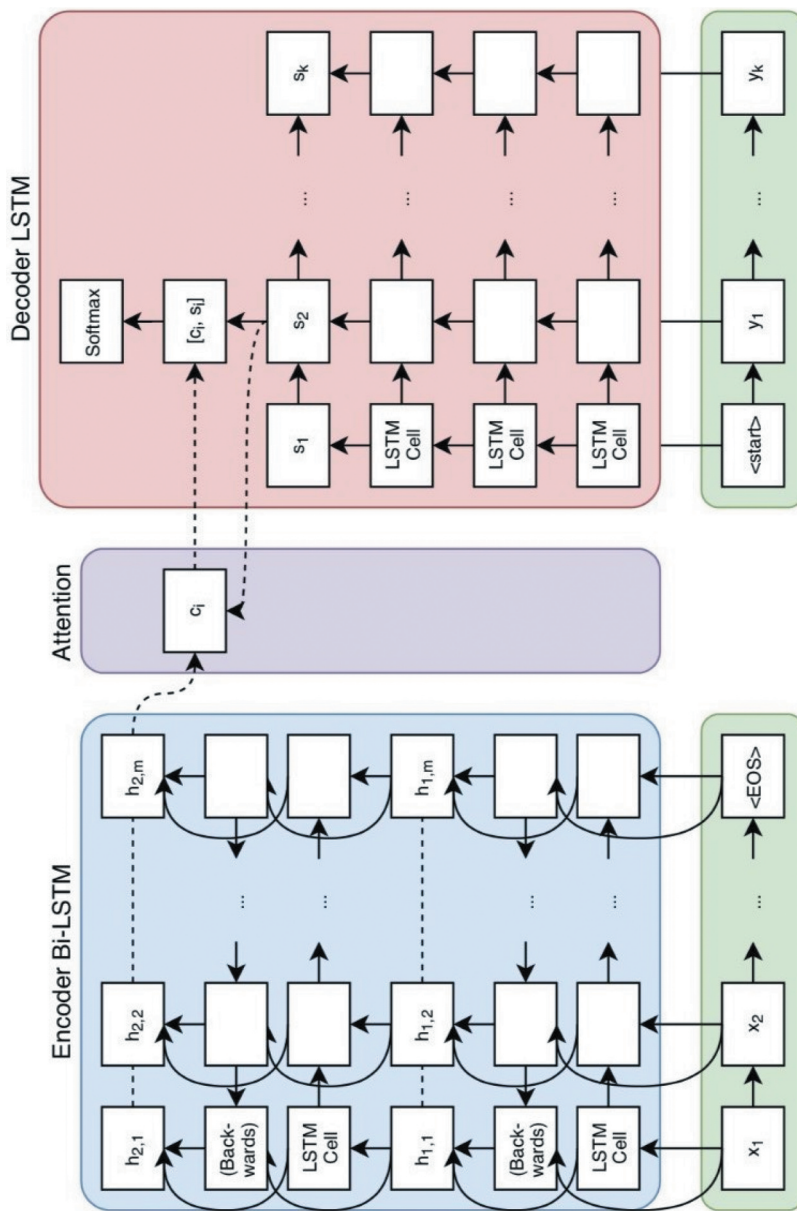


Figure 5. Encoder-Decoder architecture with an attention mechanism (Bahdanau, Cho, and Bengio 2016).

the likelihood of the sequence, based on the conditional probability of y_u , given the input feature \mathbf{h} and the previous labels $y_1 : u - 1$, using the chain rule:

$$p(\mathbf{y}|\mathbf{x}) = \prod_u p(y_u|\mathbf{h}, y_{1:u-1}) \quad (6)$$

The input to the encoder is a word embedding with $d = 768$ elements. To implement each of the encoder and decoder models, we use 4 BLSTM stack layers. The encoder model's output is a fixed-size vector that represents the internal representation of the input sequence. The number of memory cells in each layer is 256.

Hence, we use the generator network to learn a mapping function from a real word vector sample X to generated a sample y_{gen} which is corresponding to a real word vector y_{real} . The discriminator network D is a CNN network used to evaluate how well the generator network generates fake samples. The discriminator inputs all the generated vectors and tries to distinguish between the real and generated vectors.

The network's output is a 768-dimensional vector, where it is a closely aligned word vector to the model's input word vector. To learn word embedding mapping, we use an iterative refinement to find the final mapping. First, we produce the seed dictionary through the exact dictation words. Next, the system refines the dictionary until convergence. The proposed algorithm used to find the dictionary matrix D is shown below.

Input: X (source language word embeddings)

Input: Z (target language word embeddings)

Input: D (seed dictionary)

1: Until convergence:

1.1: Mapping_GAN_Model \leftarrow LEARN_MAPPING (X, Y, D)

1.2: $D \leftarrow$ LEARN_DICTIONARY ($X, Y, \text{Mapping_GAN_Model}$)

1.3: EVALUATE_DICTIONARY(D)

Output: D

We use the dot product as the similarity measure to learn a dictionary, roughly equivalent to cosine similarity between the source language word embeddings and the target language word embeddings. We set $D_{ij} = 1$ if $j = \underset{k}{\text{argmax}}(y_{\text{gen}} \text{dot} Y_{k^*})$ and for otherwise, we set $D_{ij} = 0$.

Results

To induce word embeddings, we use Wikipedia text dumps. We create independent monolingual word embeddings in each language using Wod2vec in the genism library. In our experiments, we set $d = 728$ for the number of

Table 4. Implemented model's performance in different networks.

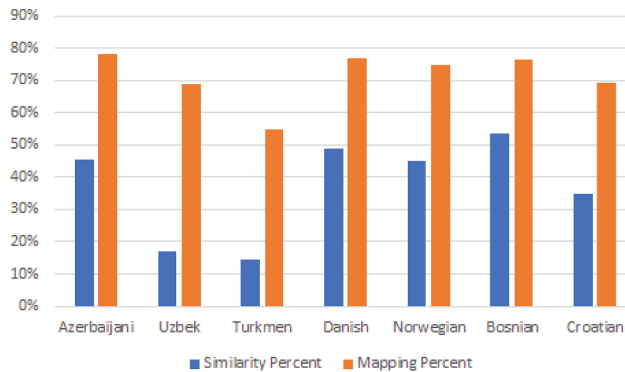
Language pairs	Vanilla LSTM	Encoder-decoder	Encoder-decoder with attention	Our proposed model
Turkish-Azerbaijani	53.2%	54.6%	55.3%	78.32%
Turkish-Uzbek	54.1%	55.2%	55.9%	68.79%
Turkish-Turkmen	47.3%	49.01%	53.65%	54.63%
Swedish-Danish	51.37%	53.6%	56.4%	77.04%
Swedish-Norwegian	50.14%	52.86%	56.2%	74.96%
Serbian-Bosnian	54.3%	55.8%	58.3%	76.48%
Serbian-Croatian	49.12%	51.16%	54.8%	69.14%
Average	51.36%	53.18%	55.79%	71.34%

dimensions of word embeddings and $w = 5$ for the size of context window. Each word embeddings vector contains floating-point numbers within the range -8 to $+8$. Experiments are conducted on the Google Colab server.

We implemented the model using TensorFlow and Keras. Backpropagation through time (BPTT) and Adam optimizer with learning rate 0.001 are used to optimize the objective function. We implemented four neural networks to find the best bilingual mapping model, including Vanilla LSTM, Encoder-decoder, Encoder-decoder with attention, and our proposed model. All of the implemented models are trained at least 1000 epochs, and the batch size is set to 500. The models take around 8–10 hours to train in the Google Colab server system, except for our proposed model, which takes approximately 11–12 hours. The similarity percentage between two vectors y_{real} and y_{gen} is computed using the following formula:

$$Sim(y_{gen}, y_{real}) = \frac{1}{1 + Euc_dis(y_{gen}, y_{real})} * 100 \quad (7)$$

The mean similarity of each language pair is obtained using the mean of all similarities in it. Table 4 summarizes the accuracy of our proposed model compared to the other implemented models. The results show that the highest performance is achieved in the proposed model. The impact of the initial

**Figure 6.** Initial seed dictionary impact on the bilingual transform mapping.

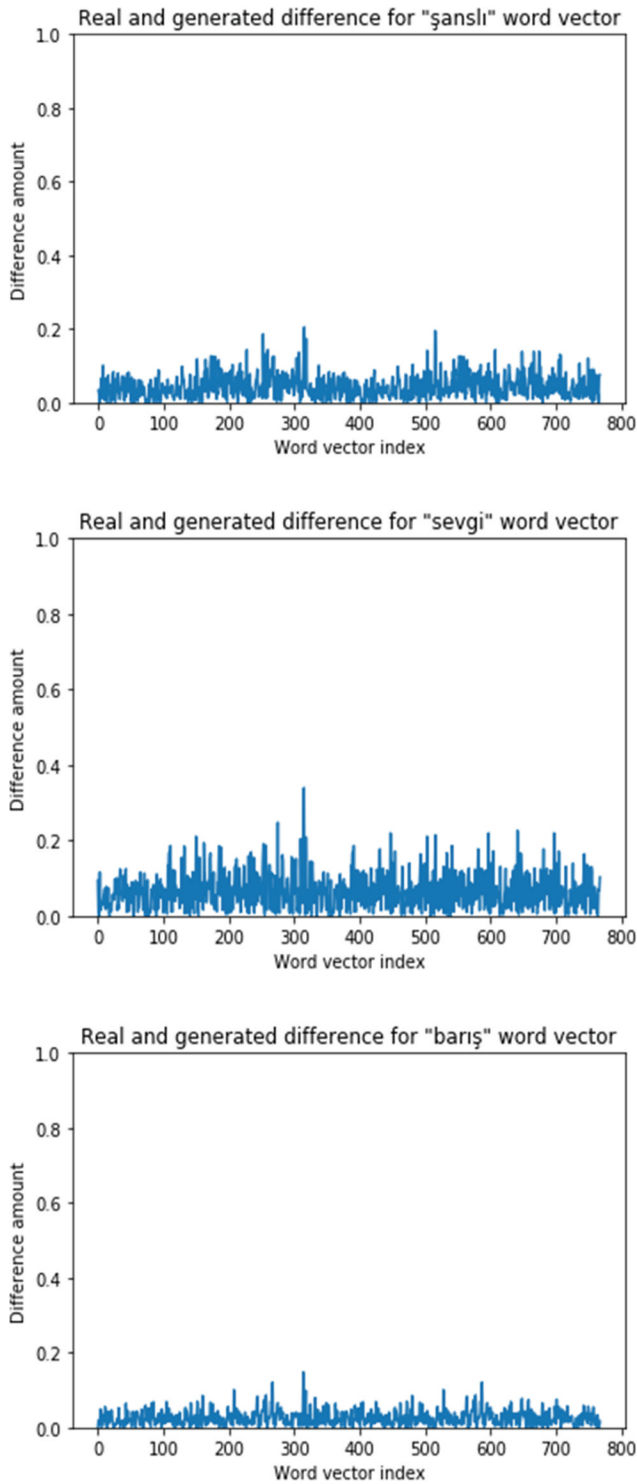


Figure 7. Differences between real and generated vectors in 3 sample words.

Table 5. Accuracy of the proposed method compared with previous works.

Work	Language pair	Accuracy
Mikolov, Le, and Sutskever (2013)	EN-DE	35.00%
Faruqui and Dyer (2014)	EN-IT	38.40%
Shigeto et al. (2015)	EN-DE	43.07%
Lazaridou, Dinu, and Baroni (2015)	EN-DE	38.93%
Lazaridou, Dinu, and Baroni (2015)	EN-IT	40.20%
Xing et al. (2015)	EN-DE	41.27%
Zhang et al. (2016)	EN-DE	40.80%
Artetxe, Labaka, and Agirre (2016)	EN-DE	41.87%
Smith et al. ()	EN-DE	43.33%
Artetxe, Labaka, and Agirre (2018)	EN-IT	45.27%
Our proposed method	TR-AZ	78.32%
Our proposed method	TR-UZ	71.79%
Our proposed method	TR-TK	54.63%
Our proposed method	SV-DA	77.04%
Our proposed method	SV-NO	74.96%
Our proposed method	SR-BS	76.48%
Our proposed method	SR-CR	69.14%
Our proposed method Average		71.77%

dictionary mass on the quality of the results is shown in Figure 6. For example, for the Azerbaijani column, we calculated the rate of its similar words by Turkish to its all words ($82/140 = 46\%$). Our experiments show that mass seed dictionaries increase the quality of mapping.

In Figure 7, we show the difference between the real and generated vectors of 3-sample word vectors (the word vectors of *şanslı*, *sevgi*, and *barış*).

Previous works have used different methods to learn bilingual word embedding mappings; Table 5 reports previous methods' best results compared to the proposed method. These results demonstrate that our method produces better mappings than previous ones.

Conclusion

This paper proposes a new method to learn bilingual word embedding mapping that improves previous works (Artetxe, Labaka, and Agirre 2016; Faruqui and Dyer 2014; Smith et al. ; Xing et al. 2015; Zhang et al. 2016). We used a GAN model to learn bilingual correspondence from monolingual corpora and initial seed dictionary. Our approach's effectiveness suggests potential NLP task applications, which require a word-level bilingual transfer, such as bilingual machine translation.

Notes

1. <https://github.com/RaRe-Technologies/gensim>.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

ORCID

Ghafour Alipour  <http://orcid.org/0000-0002-2070-4334>

References

- Ammar, W., G. Mulcaire, Y. Tsvetkov, G. Lample, C. Dyer, and N. A. Smith. 2016. Massively multilingual word embeddings. CoRR abs/1602.01925. <http://arxiv.org/abs/1602.01925>
- Arjovsky, M., S. Chintala, and L. Bottou. 2017. Wasserstein GAN. *CORR* abs/1701.07875
- Artetxe, M., G. Labaka, and E. Agirre. 2016. Learning principled bilingual mappings of word embedding while preserving monolingual invariance. Conference on empirical methods in natural language processing, 2289–94 .
- Artetxe, M., G. Labaka, and E. Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. Proceedings of ACL, ACL, 451–62.
- Artetxe, M., G. Labaka, and E. Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations, Proceedings of the AAAI Conference on Artificial Intelligence, 32 1 <https://ojs.aaai.org/index.php/AAAI/article/view/11992>
- Bahdanau, D., K. Cho, and Y. Bengio. 2016. Neural Machine Translation by Jointly Learning to Align and Translate. <https://arxiv.org/abs/1409.0473>
- Bojanowsk, P., E. Grave, A. Joulin, and T. Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–46. doi:10.1162/tacl_a_00051.
- Collobert, R., and J. Weston. 2008. A unified architecture for natural language processing. Proceedings of the 25th International Conference on Machine Learning - ICML '08. 20 (1) 160–167.
- Conneau, A., G. Lample, M. Ranzato, L. Denoyer, and H. Jégo. 2018. Word translation without parallel data, 6th International Conference on Learning Representations Vancouver, BC, Canada, OpenReview.net.
- Dinu, Georgiana, Lazaridou, Angeliki, and Baroni, Marco. 2015. Improving zero-shot learning by mitigating the hubness problem, In Proceedings of ICLR (Workshop Track).
- Duong, L., H. Kanayama, T. Ma, S. Bird, and T. Cohn. 2016. Learning cross-lingual word embeddings without bilingual corpora, Proceedings of EMNLP, 1285–1295.
- Faruqui, M., and C. Dyer. 2014. Improving vector space word representations using multi-lingual correlation. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, 462–71.
- Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, and S. Ozair. 2014. Generative adversarial nets. *Neural Information Processing Systems* 27 2672–2680.
- Gouws, S., Y. Bengio, and G. Corrado. 2015. Fast bilingual distributed representations without word alignments. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), Red Hook, NY, 748–756.
- Hammarström, H., R. Forkel, and M. Haspelmath. 2017. Turkic . In *Glottolog 3.0.*,” Jena, Germany: Max Planck Institute for the Science of Human History, vol. 3.
- Hauer, Bradley, Garrett, Nicolai, and Grzeg, Kondrak. 2017. Bootstrapping unsupervised bilingual lexicon induction. In Proceedings of EACL, 619–624

- Hoshen, Y., and L. Wolf. 2018. Non-adversarial unsupervised word translation. Proc. of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Eight Street, Stroudsburg, PA, 18360 United States, 469–78.
- Iacer, Calixto, Qun, Liu, and Nick, Campbell. 2017. Multilingual Multi-modal Embeddings for Natural Language Processing, CoRR, abs/1702.01101
- Jinsong, S., S. Zhenqiao, L. Yaojie, X. Mu, W. Changxing, and C. Yidong. 2018b. Exploring implicit semantic constraints for bilingual word embeddings. *Neural Process Letter* 48 1073–1088. doi: <https://doi.org/10.1007/s11063-017-9762-8>
- Jinsong, S., W. Shan, Z. Biao, W. Changxing, Q. Yue, and X. Deyi. 2018a. A neural generative autoencoder for bilingual word embeddings. *Information Sciences* 424 287–300. doi: [10.1016/j.ins.2017.09.070](https://doi.org/10.1016/j.ins.2017.09.070)
- Joulin, A., E. Grave, P. Bojanowski, and T. Mikolov. 2016. Bag of tricks for efficient text classification. <https://arxiv.org/abs/1607.01759v1>
- Kondrak, G., B. Hauer, and G. Nicolai. 2017. Bootstrapping unsupervised bilingual lexicon induction. Proceedings of EACL 2 , 619–624. doi: [10.18653/v1/E17-2098](https://doi.org/10.18653/v1/E17-2098).
- Lample, G., A. Conneau, L. Denoyer, and M. Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. [arXiv:1711.00043](https://arxiv.org/abs/1711.00043)
- Lazaridou, A., G. Dinu, and M. Baroni. 2015. Hubness and pollution: Delving into cross space mapping for zero-shot learning. Proceedings of ACL, Beijing, China.
- Levy, O., A. Søgaard, and Y. Goldberg. 2017. A strong baseline for learning cross-lingual word embeddings from sentence alignments. *Proceeding of EACL* 1 765–774.
- Levy, O., Y. Goldberg, and I. Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3:211–25. doi: [10.1162/tacl_a_00134](https://doi.org/10.1162/tacl_a_00134).
- Lu, Ang, Wang, Weiran, Bansal, Mohit, Gimple, Kevin, and Livescu, Karen. 2015. Deep multilingual correlation for improved word embeddings Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 1 250–256.
- Luong, M., and C. Manning. 2016. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. <https://arxiv.org/abs/1604.00788>
- Luong, T., H. Pham, and C. D. Manning. 2015. Bilingual word representations with monolingual quality in mind. Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, 151–59 doi: [10.3115/v1/W15-1521](https://doi.org/10.3115/v1/W15-1521).
- Makhzani, A., J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. 2016. Adversarial autoencoders. <https://arxiv.org/abs/1511.05644>
- Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013b. Distributed representations of words and phrases and their compositionality. <https://arxiv.org/abs/1310.4546>
- Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv:1301.3781* 2:3111–19.
- Mikolov, T., Q. V. Le, and I. Sutskever. 2013. Exploiting similarities among languages for machine translation. <https://arxiv.org/abs/1309.4168>
- Mogadala, Aditya, and Rettinger, Achim. 2016. Bilingual word embeddings from parallel and nonparallel corpora for cross-language text classification Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 692–702.
- Mrkšić, N., I. Vulić, D. Ó. Séaghdha, Í. Leviant, R. Reichart, M. Gašić, and A. Korh. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics* 5:309–24. doi: [10.1162/tacl_a_00063](https://doi.org/10.1162/tacl_a_00063).

- Pennington, J., R. Socher, and C. Manning. 2014. Glove: Global vectors for word representation. D14-1162 2014 Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) Doha, Qatar (Association for Computational Linguistics), 1532–1543 <https://aclanthology.org/D14-1162> doi:10.3115/v1/D14-1162.
- Rajendran, Janarthanan, Khapra, Mitesh M, Chandar, Sarath, and Ravindran, Balaraman. 2016. Bridge correlational neural networks for multilingual multimodal representation learning, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 171–181
- Ruder, S., I. Vulic, and A. Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research* 65:569–631. doi:10.1613/jair.1.11640.
- Shigeto, Y., I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto. 2015. Ridge Regression, Hubness, and Zero-Shot Learning. <https://arxiv.org/abs/1507.00825>
- Smith, S. L., D. H. Turban, S. Hamblin, and N. Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. 5th International Conference on Learning Representations (ICLR 2017), April 24-26 2017 (OpenReview.net) Toulon, France.
- Upadhyay, S., M. Faruqui, C. Dyer, and D. Ro. Cross-lingual models of word embeddings: An empirical comparison. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Vol. 1. (Long Papers).
- Valerio, A., and M. Barone. 2016. Towards crosslingual distributed representations without parallel text trained with adversarial autoencoders. Proceedings of the 1st Workshop on Representation Learning for NLP Berlin, Germany (Association for Computational Linguistics), 121–126 <https://aclanthology.org/W16-16> doi:10.18653/v1/W16-16.
- Vulić, I., and M.-F. Moens. 2015. Bilingual word embeddings from non-parallel DocumentAligned data applied to bilingual lexicon induction. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China. (Association for Computational Linguistics), 719–725. <https://aclanthology.org/P15-2> doi:10.3115/v1/P15-2.
- Vulić, I., and A. Korhonen. 2016. On the role of seed lexicons in learning bilingual word embeddings. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, (Association for Computational Linguistics), 247–57 <https://aclanthology.org/P16-1> doi:10.18653/v1/P16-1.
- Xing, C., D. Wang, C. Liu, and Y. Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. Proceedings of NAACL-HLT Denver, USA (Association for Computational Linguistics), 1005–10.
- Zeman, D., J. Hajič, M. Popel, M. Potthast, M. Straka, F. Ginter, ... S. Petrov. 2018. Multilingual parsing from raw text to universal dependencies. In Proceedings of the CoNLL 2017 Shared Task Vancouver, Canada (Association for Computational Linguistics), 1–19 <https://aclanthology.org/K17-3> doi:10.18653/v1/K17-3.
- Zhang, M., Y. Liu, H. Luan, and M. Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics Vancouver, Canada, Vol. 1 (Association for Computational Linguistics), 1959–1970 <https://aclanthology.org/P17-1> doi:10.18653/v1/P17-1.
- Zhang, Y., D. Gaddy, R. Barzilay, and T. Jaakkola. 2016. Ten pairs to tag – multilingual POS tagging via coarse mapping between embeddings. Proceedings of NAACL-HLT San Diego, USA (Association for Computational Linguistics), 1307–1317.
- Zhang, Y., Y. Li, Y. Zhu, and X. Hu. 2020. Wasserstein GAN based on Autoencoder with back-translation for cross-lingual embedding mappings. *Pattern Recognition Letters* 129:311–16. doi:10.1016/j.patrec.2019.11.033.