Iterative training for unsupervised word embedding mapping

Sijia Yu and Yves Lepage

Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
ysy@fuji.waseda.jp, yves.lepage@waseda.jp

Abstract

The problem of word embedding mapping is to map word embeddings trained independently for two languages one onto another, i.e., to map each word in one language onto a word in the other language. We propose an iterative training of the state-of-the-art method but we modify the loss function of the generator of the GAN used. In addition, we introduce an iterative adjustment method to produce a pseudo-dictionary which expands at each iteration while minimising the average Euclidean distance between source and target words in the pseudo-dictionary. We present a range of experiments with several variants. In our experiments, we obtained slight but consistent improvements over the state-of-the art method. We provide measures of correlation with machine translation scores for all the variants.

1. Introduction

Word embeddings are continuous representations of words in high dimension vector spaces built using various models, like the CBOW model (Mikolov et al., 2013a). Previous work (Mikolov et al., 2013b) showed that the distribution of words with the same meaning in different vector spaces built independently for different languages, exhibit similar geometric structures in reduced dimensional spaces obtained using dimension reduction methods like principal component analysis (PCA). This led to the intuition that word embeddings for different languages could be linearly mapped one onto another.

The work in bilingual word embedding mapping started from *supervised* learning with thousands of pairs of words in the dictionary. It can be considered as minimising the least square loss or squared Euclidean distance between the mapping word embedding and the target word embedding models (Mikolov et al., 2013b).

$$\min \sum_{i=1}^{n} \|Wx_i - z_i\|^2 \tag{1}$$

In Equation (1), $W \in \mathbb{R}^{d \times d}$ is the mapping matrix, $(x_i, y_i)_{i \in 1,2 \cdots n}$ are pairs of word vectors from the source and target matrices $X, Z \in \mathbb{R}^{n \times d}$ of each embedding model. The problem can be treated as a quadratic problem by constraining $W \in R^{d \times d}$ to be an orthogonal matrix and by normalising the word embedding vectors in length (Xing et al., 2015). Using singular value decomposition (SVD) can be used to find an orthogonal mapping matrix W (Artetxe et al., 2016). Recent work showed that *unsupervised* methods can also be used to map word embeddings separately trained from monolingual corpora (Conneau et al., 2017; Artetxe et al., 2018)

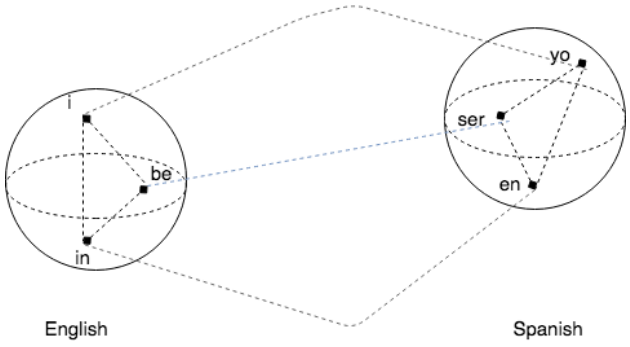

Figure 1: Illustration of bilingual word embedding mapping from English (*left*) to Spanish (*right*).

The contribution of this paper is as follows. We apply iterative training with the state-of-the-art generative adversarial neural networks (GAN) model for unsupervised learning of word embedding mappings with a modified generator loss function to ensure better back translation. We also apply an iterative adjustment of the mapping which relies on iteratively updating a pseudo-dictionary used for internal evaluation.

The paper is structured as follows. Section 2. reviews the main techniques used in the state-of-the-art techniques. Section 3. describes our proposal, i.e., iterative training and iterative adjustment, with some improvements. Section 4. describes the data used in the experiments, the settings and the results obtained over a range of models.

2. Bilingual word embedding mapping

2.1. Orthogonal mapping matrix and singular value decomposition

It was showed that normalising word vectors in length and applying an orthogonal transformation makes the mapping matrix of a higher quality (Xing et al., 2015). Normalising the matrix can be formulated as the minimalisation problem (Artetxe et al., 2016), shown in (2).

$$W = \arg\min_{W} \|WX - Z\|_F, W^T W = I \tag{2}$$

Here, W is the mapping matrix and X, Z are the source and target word embedding matrices. Our goal is to minimise the Frobenius norm $\|WX - Z\|_F$ between WX and Z, in simpler words, minimising the square error rate between the two matrices WX and Z, and to constrain W to be orthogonal. This can be achieved by applying singular value decomposition (SVD) on $Z^T X$ to obtain $U \Sigma V^T$ where $W = UV^T$. The objective is written as in (3).

$$
\begin{aligned}
W &= \arg\min_W \sum_i \|Wx_i - z_i\|^2 \\
&= \arg\min_W \sum_i \left(\|Wx_i\|^2 + \|z_i\|^2 - 2 \times \langle Wx_i, z_i \rangle \right) \\
&= \arg\max_W \sum_i \langle Wx_i, z_i \rangle
\end{aligned} \tag{3}
$$

where $\langle ., . \rangle$ notes the inner product. The objective can be achieved by maximising the trace of the matrix, i.e., by looking for a matrix W such that $W = \arg\max_W Tr(WXZ^T)$, with $Z^TX = U\Sigma V^T$. This is equivalent to determine $W = \arg\max_W Tr(\Sigma^T U^T W V)$. Since U and V are orthogonal and since Σ is diagonal, $W = UV^T$ maximises the trace.

2.2. Search methods

2.2.1. Nearest neighbour (NN) search

Given a source word, the nearest neighbour (NN) search method retrieves the target word vector which is ranked the first to the source word vector by similarity.

$$
\text{NN}(Wx_i, Z) = \arg_{z_j \in Z} \min \cos(Wx_i, z_j) \tag{4}
$$

In practice, when retrieving target words from source words using NN search, some target words exhibit a tendency to be near many source word vectors at the same time. They are wrongly considered to be the translation of too many source words, i.e., they attract too many words. For that reason they are called *hubs*, and their existence is called the *hubness problem* (Dinu and Baroni, 2014).

2.2.2. Cross domain similarity local scaling (CSLS)

To address the hubness problem, a method called cross domain similarity local scaling (CSLS) was introduced in (Conneau et al., 2017). For a pair of words (x_i, w_j), the objective is to minimise the distance between the centroid of the k nearest neighbours of x_i mapped onto the target domain and the centroid of the k nearest neighbours of z_j mapped back onto the source domain. It thus relies on computing the average similarity between Wx_i and the centroid of the set of its closest target neighbours obtained through mapping:

$$
M_t(Wx_i) = \frac{1}{k} \sum_{z_j \in T_k(Wx_i)} \cos(Wx_i, z_j) \tag{5}
$$

$T_k(Wx_i)$ is the set of the k nearest neighbours to Wx_i in the target domain. $M_s(z_j)$ can be computed in the other direction in a similar way from a target word z_j.

For a given vector Wx_i obtained by mapping a word x_i onto the target domain, the CSLS method will thus retrieve the word z_j which will minimise the criterion given in Equation (6).

$$
\begin{aligned}
\text{csls}(Wx_i, z_j) = 2\cos(Wx_i, z_j) \\
- M_t(Wx_i) - M_s(z_j)
\end{aligned} \tag{6}
$$

This criterion should penalise hubs, and thus mitigate the hubness problem.

2.3. Unsupervised bilingual embedding mapping: use of GANs

State-of-the-art methods for solving the word embedding mapping problem in its unsupervised version, use generative adversarial neural networks (GANs). The loss functions for the discriminator and the generator are defined respectively in Equations (7) and (8).

$$
\begin{aligned}
L_D(\theta_D|W) = &- \frac{1}{n} \sum_{i=1}^{l} \log(P_D(Wx_i)) \\
&- \frac{1}{m} \sum_{j=1}^{m} \log(1 - P_D(z_j))
\end{aligned} \tag{7}
$$

$$
\begin{aligned}
L_W(W|\theta_D) = &- \frac{1}{m} \sum_{j=1}^{m} \log(P_D(z_j)) \\
&- \frac{1}{n} \sum_{i=1}^{n} \log(1 - P_D(Wx_i))
\end{aligned} \tag{8}
$$

The generator is simply the linear mapping matrix W. The problem is considered a binary classification problem, hence, the output of the discriminator, P_D, is a number between 0 to 1. Its parameters are θ_D; n and m are the number of samples in the mapped embedding and the target embedding respectively.

As for the loss of the discriminator, in the ideal situation, $P_D(Wx_i)$ should be 1 and $P_D(z_j)$ should be 0, meaning that the discriminator correctly distinguishes between the source and target word vectors. In contrast, the generator tries to fool the discriminator to judge the embeddings in an opposite way. The complete procedure is that the generator first generates some mapping embeddings from source embeddings randomly, and the discriminator judges whether the word embedding vectors are from the source embedding space or the target embedding space, according to which the discriminator and the generator update their parameters respectively.

Algorithm 1 Procrustes refinement

Require:
Source embedding: X
Target embedding: Z
Input mapping matrix: W
Ensure:
repeat
$F \leftarrow$ CSLS retrieval by (X,Z,W)
$W \leftarrow VU^T$, $X^TFZ = U\Sigma V^T$
until convergence
return W

2.4. Procrustes refinement

The Procrustes refinement (Artetxe et al., 2017; Conneau et al., 2017) imposes the orthogonality constraint on a bilingual embedding mapping matrix W and minimises its Frobenius norm at the same time. This refinement is implemented by the self-learning algorithm proposed in (Artetxe et al., 2017; Conneau et al., 2017). It consists in building a matrix F by using CSLS retrieval from an initial matrix W.

The orthogonality constraint is applied on this matrix by using SVD, as described in Section 2.1. to get a new matrix W. This is repeated until convergence (see Algo. 1).

2.5. Model selection

A problem in unsupervised word embedding mapping is to find a validation to select a best model W. The validation criterion used in (Conneau et al., 2017) calculates the average cosine similarity between those word vectors in a pseudo-dictionary $\mathcal{D}$ retrieved by CSLS in the most 10,000 frequent words. The model W which is selected as output is the one which maximises the average cosine similarity between the word vectors in the pseudo-dictionary.

3. Proposal: iterative training and adjustment

We propose to make the training procedure of bilingual embedding mapping more precise by applying the three following processes in the following order.

- Firstly we perform an iterative training using a GAN. This is described in Algorithm 2. However, different from previous work, we add a term in the loss function of the generator.
- Secondly we perform an iterative adjustment of the obtained mapping. It is the same as the first step, except that no GAN is used here. Algorithm 2 is executed again, but without the line marked ♠. During this iterative process, the mapping matrix W is adjusted because the pseudo-dictionary changes at each iteration.
- Thirdly, we apply Procrustes refinement as described in Section 2.4., and as is the case in previous works.

The improvements introduced above are presented in each of the following sub-sections.

Algorithm 2 Iterative training using GAN (with line ♠) or iterative adjustment (without line ♠)

Require:
  Source embedding: X
  Target embedding: Z
  A diagonal mapping matrix W
    for initialisation in the case of
      iterative training with GAN
    from iterative training in the case of
      iterative adjustment
Ensure:
  repeat
    Determine $\mathcal{D}$ (see Section 3.2.)
    ♠ Train using GAN (see Section 2.3.)
      with new generator loss (see Section 3.1.)
    Impose orthogonality constraint, i.e.,
      update W using Eq. (11) (see Section 3.3.)
    Compute average cosine similarity
      over word pairs in $\mathcal{D}$
  until average cos. sim. decreases (see Section 2.5.)
  return W

3.1. Modification of the loss function

The first improvement we bring is in the assessment of the quality of the mapping while learning W with GAN. A direct measure of the Euclidian distance between the source words in the pseudo-dictionary $\mathcal{D}$ and their estimated translations gives an idea of how close they are: the smaller the average Euclidian distance, the more exact the bilingual embedding mapping. Globally, we aim at minimising the average Euclidean distance over all the words in $\mathcal{D}$. We thus add the term given in Equation (9) to the loss.

$$L_G = -\frac{1}{k}\sum_{i=1}^{k} \|Wx_i - z_i\|^2 \tag{9}$$

k is the size of the pseudo-dictionary. The generator loss function we use in our GAN is $L_W(W|\theta_D)$ as defined in (8), plus the term L_G in Equation (9).

3.2. Pseudo-dictionary for iterative training and iterative adjustment

The second improvement we bring is in the selection of the words present in a pseudo-dictionary $\mathcal{D}$ which is used to internally assess the quality of the mapping. In the state-of-the-art method, there is also a pseudo-dictionary, but the words are selected at random. We think that the words in $\mathcal{D}$ should be selected according to their probability of being in an actual dictionary between the two languages at hand as determined by the current mapping. As such a dictionary is bidirectional, we select the word pairs from the source language to the target language, and back. At each iteration, we take all the common word pairs by retrieving X and Z using the CSLS search method, in both directions, from X to Z and from Z to X. $\mathcal{D}$ is thus defined as the intersection of the two sets of pairs of words retrieved using the CSLS search method in both direction, as shown in (10).

$$\mathcal{D} \leftarrow \text{CSLS}(WX, Z) \cap \text{CSLS}(Z, WX) \tag{10}$$

If the overall training is able to learn a better mapping, the number of word pairs should increase, hence, the pseudo-dictionary should expand. Simultaneously, the value of the generator loss should decrease when used in conjunction with the GAN. When used in the iterative adjustment process, the changes in the pseudo-dictionary has the effect of refining and better adjusting the mapping.

3.3. Ensuring an orthogonal mapping

So as to ensure an orthogonal mapping we use the technique presented in (Cisse et al., 2017). It consists in updating the matrix W using the formula in (11).

$$W \leftarrow (1+\beta)W - \beta WW^TW \tag{11}$$

The overall iterative training is stopped by using the same criterion as in the state-of-the-art technique, i.e., the model selected at the end is selected as presented in Section 2.5. This is also used in the state-of-the-art method that we use for our baseline.

Parameter	Value
β for orthogonal learning	0.001
Batch size	32
Discriminator	
Number of layers	2
Number of layer nodes	2,048
Dropout	0.1
Learning rate decay	0.98

Table 1: Parameter values for bilingual embedding mapping experiments

4. Experiments

4.1. Languages and data sets

We test and compare our proposal on six European languages: German (de), Spanish (es), English (en), Finnish (fi), French (fr) and Italian (it); more precisely, on all language pairs involving English: en–es, en–de, en–fi, en–fr and en–it, in both directions, This makes a total of 10 language pairs.

The reason to choose these language pairs lies in the availability of comparable data. Firstly, freely pretrained word embedding models for all these languages are available (Grave et al., 2018; Bojanowski et al., 2017).[1] Secondly, a freely aligned multilingual corpus involving all these languages is also available, the Europarl corpus (Koehn, 2005).[2]

As for the comparability of data, the word embeddings were all trained on Wikipedia, known to be comparable across at least European languages. The dimension of the embedding space is 300 for all languages.

The Europarl corpus in its version 3 contains a relatively large amount of lines common to all the 6 chosen languages. It is thus possible to run translation experiments on corresponding data, so that the accuracy reported can be supposed to reflect linguistic differences across the language pairs considered. In our translation experiments, we used 480,000 lines of the Europarl corpus v.3, aligned across all the selected languages. One tenth, i.e., 48,000 lines is used for testing. the other nine tenths are used for training and development, As usual, BLEU is used to measure translation accuracy.

4.2. Experimental settings

The experiments on bilingual embedding mapping involve some parameter settings given in Table 1.

We first applied our iterative GAN with the most frequent 75,000 words, for iterative adjustment, we used the most 20,000 frequent words and for Procrustes refinement, the most 15,000 frequent words. We performed iterative adjustment 30 times and the Procrustes refinement 5 times.

As for machine translation experiments, we used the standard GIZA++/MOSES SMT system (Koehn, Philipp et al., 2007) with its default settings.

[1] `https://fasttext.cc/docs/en/crawl-vectors.html`

[2] `http://www.statmt.org/europarl/`

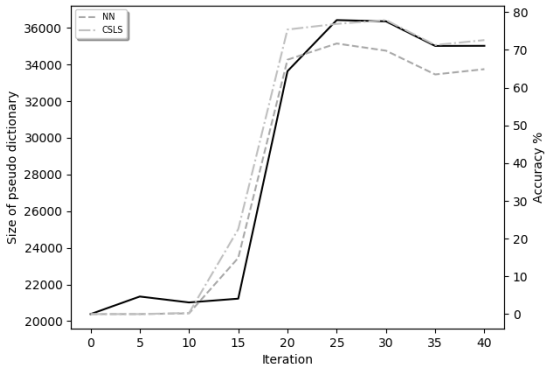

Figure 2: The dotted curves give the word embedding mapping accuracy along iterative training every 5 iterations (en–es), computed before Procrustes refinement. The black curve is the size of the pseudo-dictionary.

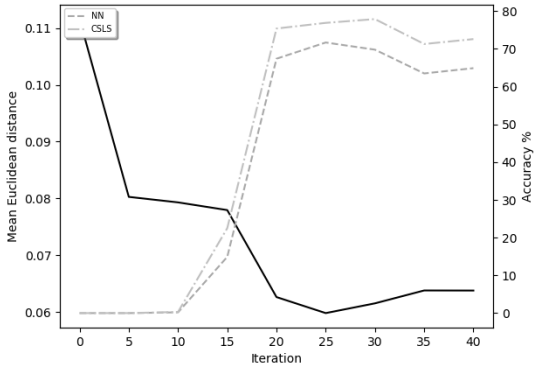

Figure 3: The dotted curves give the word embedding mapping accuracy along iterative training every 5 iterations (en–es), computed before Procrustes refinement. The black curve is the average Euclidian distance (Eq. (9)).

4.3. Experiment results

The accuracy of the methods measured using the reference dictionaries released in (Conneau et al., 2017) are given in Table 2. In the best configurations for the (GAN CSLS +pro) and our method (Ours CSLS +adj+pro), our method consistently slightly outperforms the state-of-the-art method, in all language pairs, except for English–German.

The addition of the Procrustes refinement is always positive, even in the case of the addition of our iterative adjustment (compare line Ours +pro with line Ours +adj+pro in Table 2).

When comparing searching methods, CSLS is consistently better than the NN search method, as well as with our method. This confirms the hypothesis that the hubness problem is reduced by CSLS.

The correlation between accuracy of bilingual embedding mapping and translation accuracy is relatively high across all methods. It does not vary when comparing the state-of-the-art method with our method in their basic or refined versions. However it consistently increases for re-

	Search	Refinement	en-de	de-en	en-es	es-en	en-fi	fi-en	en-fr	fr-en	en-it	it-en	Pearson's corr.
GAN	NN		61.0	51.4	65.0	67.5	12.7	21.7	66.5	63.0	55.9	55.7	0.74
		+pro	71.0	69.9	78.4	79.3	36.8	56.9	77.9	77.9	74.5	74.9	0.84
	CSLS		68.9	59.4	74.8	76.3	21.1	28.3	75.3	71.1	65.3	64.3	0.75
		+pro	**75.1**	72.9	**82.3**	84.2	43.7	59.8	81.7	82.3	77.8	78.1	0.84
Ours	NN		63.3	64.3	71.7	71.1	21.6	32.1	67.9	63.1	63.1	64.9	0.78
		+adj	69.0	69.1	78.9	78.9	33.0	51.5	77.5	76.4	73.5	73.3	0.84
		+adj+pro	71.0	69.5	79.8	79.5	38.2	57.3	78.3	78.3	75.3	75.3	0.84
	CSLS		70.1	68.9	76.9	78.5	31.7	42.3	77.7	72.9	73.5	73.3	0.78
		+adj	73.7	73.1	82.1	83.7	41.8	56.6	81.6	81.4	77.8	77.8	0.84
		+adj+pro	74.5	**73.3**	**82.3**	**84.5**	**45.7**	**61.3**	**82.5**	**82.7**	**78.3**	**78.3**	0.85
BLEU scores			34.1	34.7	19.9	27.1	29.4	29.4	25.6	31.0	14.6	23.7	

Table 2: Accuracy for the state-of-the-art (GAN) and our iterative training method (ours), in several variants, i.e., using both search methods (NN or CSLS) and with or without refinement methods (+adj for iterative adjustment and +pro for Procrustes refinement). Boldface numbers are the best scores on each column. The last line in the table gives the BLEU scores. The last column gives Pearson's correlation of the accuracy with the BLEU scores over all language pairs (all p-values were less than 1×10^{-2}).

fined versions, which is a positive point, indicating that the translation dictionaries obtained from word embedding mappings obtained by refined versions could be more useful in a machine translation systems. However, the small number of language pairs used here to compute the correlation may be questioned. In future work, experiments on all available Europarl languages could be considered to get more reliable correlation values.

Figure 2 traces the size of the pseudo-dictionary output by the iterative adjustment process and the accuracy for the English–Spanish language pair along iterations. The curves fit one to another for both search methods (NN and CSLS). This shows a strong relationship between the size of the pseudo-dictionary and the accuracy of the model.

Figure 3 traces the accuracy of the model and the average Euclidian distance along the iterations for the same language pair along iterative training, again for both search methods (NN and CSLS). Following expectation, when the average Euclidian distance decreases, the accuracy increases. The best model is reached at iteration 25.

5. Conclusion

In this paper, we proposed an iterative training method for the unsupervised word embedding mapping problem based on the state-of-the-art method that uses a generative adversarial neural network. Firstly we perform iterative training using a modified GAN where we added a new term in the generator loss function. Secondly we perform an iterative adjustment of the mapping obtained by training using a pseudo-dictionary which is refined at each iteration to contain more and more better aligned words. Thirdly we apply the Procrustes refinement as is done in previous works. In addition, we also use a more effective orthogonalisation technique which we apply on the matrix representing the mapping. Our method consistently slightly out-performs the state-of-the-art method.

6. References

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre, 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *EMNLP 2016*.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre, 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *ACL 2017*, volume 1.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre, 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *CoRR*, abs/1805.06297.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov, 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Cisse, Moustapha, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier, 2017. Parseval networks: Improving robustness to adversarial examples. In *PMLR 2017*, volume 70.

Conneau, Alexis, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou, 2017. Word translation without parallel data. *CoRR*, abs/1710.04087.

Dinu, Georgiana and Marco Baroni, 2014. Improving zero-shot learning by mitigating the hubness problem. *CoRR*, abs/1412.6568.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov, 2018. Learning word vectors for 157 languages. In *LREC 2018*.

Koehn, Philipp, 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*.

Koehn, Philipp et al., 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007*.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean, 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever, 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Xing, Chao, Dong Wang, Chao Liu, and Yiye Lin, 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *NAACL-HLT 2015*.