Multi-Adversarial Learning for Cross-Lingual Word Embeddings

Haozhou Wang[1], **James Henderson**[2], **Paola Merlo**[1]
[1] University of Geneva
[2] Idiap Research Institute
{haozhou.wang, paola.merlo}@unige.ch, james.henderson@idiap.ch

Abstract

Generative adversarial networks (GANs) have succeeded in inducing cross-lingual word embeddings —maps of matching words across languages— without supervision. Despite these successes, GANs' performance for the difficult case of distant languages is still not satisfactory. These limitations have been explained by GANs' incorrect assumption that source and target embedding spaces are related by a single linear mapping and are approximately isomorphic. We assume instead that, especially across distant languages, the mapping is only piece-wise linear, and propose a multi-adversarial learning method. This novel method induces the seed cross-lingual dictionary through multiple mappings, each induced to fit the mapping for one subspace. Our experiments on unsupervised bilingual lexicon induction and cross-lingual document classification show that this method improves performance over previous single-mapping methods, especially for distant languages.

1 Introduction and background

Word embeddings, continuous vectorial representations of words, have become a fundamental initial step in many natural language processing (NLP) tasks for many languages. In recent years, their cross-lingual counterpart, cross-lingual word embeddings (CLWE) —maps of matching words across languages— have been shown to be useful in many important cross-lingual transfer and modeling tasks such as machine translation, cross-lingual document classification and zero-shot dependency parsing (Klementiev et al., 2012; Zou et al., 2013; Guo et al., 2015; Conneau et al., 2018; Glavaš et al., 2019; Zhang et al., 2020).

In these representations, matching words across different languages are represented by similar vectors. Following the observation of Mikolov et al. (2013) that the geometric positions of similar words in two embedding spaces of different languages appear to be related by a linear relation, the most common method aims to map between two pre-trained monolingual embedding spaces by learning a single linear transformation matrix. Due to its simple structure design and competitive performance, this approach has become the mainstream of learning CLWE (Glavaš et al., 2019; Vulić et al., 2019; Ruder et al., 2019).

Initially, the linear mapping was learned by minimizing the distances between the source and target words in a seed dictionary. Early work from Mikolov et al. (2013) uses a seed dictionary of five-thousand word pairs. Since then, the size of the seed dictionary has been gradually reduced, from several-thousand to fifty word pairs (Smith et al., 2017), reaching a minimal version of only sharing numerals (Artetxe et al., 2017).

More recent works on unsupervised learning have shown that mappings across embedding spaces can also be learned without any bilingual evidence (Barone, 2016; Zhang et al., 2017; Conneau et al., 2018; Hoshen and Wolf, 2018; Alvarez-Melis and Jaakkola, 2018; Artetxe et al., 2018). More concretely, these fully unsupervised methods usually consist of two main steps (Hartmann et al., 2019): an unsupervised step which aims to induce the seed dictionary by matching the source and target distributions, and then a pseudo-supervised refinement step based on this seed dictionary.

The system proposed by Conneau et al. (2018) can be considered the first successful unsupervised system for learning CLWE. They first use generative adversarial networks (GANs) to learn a single linear mapping to induce the seed dictionary, followed by the Procrustes Analysis (Schönemann, 1966) to refine the linear mapping based on the induced seed dictionary. While this GAN-based model has competitive or even better performance compared to supervised methods on typologically-similar language pairs, it often exhibits poor performance on typologically-distant language pairs,

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 463–472
June 6–11, 2021.

pairs of languages that differ drastically in word forms, morphology, word order and other properties that determine how similar the lexicon of a language is. More specifically, their initial linear mapping often fails to induce the seed dictionary for distant language pairs (Vulić et al., 2019). Later work from Artetxe et al. (2018) has proposed an unsupervised self-learning framework to make the unsupervised CLWE learning more robust. Their system uses similarity distribution matching to induce the seed dictionary and stochastic dictionary induction to refine the mapping iteratively. The final CLWE learned by their system performs better than the GAN-based system. However, their advantage appears to come from the iterative refinement with stochastic dictionary induction, according to Hartmann et al. (2019). If we only consider the performance of a model induced only with distribution matching, GAN-based models perform much better. This brings us to our first conclusions, that a GAN-based model is preferable for seed dictionary induction.

Fully unsupervised mapping-based methods to learn CLWE rely on the strong assumption that monolingual word embedding spaces are isomorphic or near-isomorphic, but this assumption is not fulfilled in practice, especially for distant language pairs (Søgaard et al., 2018). Experiments by Vulić et al. (2020) also demonstrate that the lack of isomorphism does not arise only because of the typological distance among languages, but it also depends on the quality of the monolingual embedding space. If we replace the seed dictionary learned by an unsupervised distribution matching method with a pretrained dictionary, keeping constant the refinement technique, the final system becomes more robust (Vulić et al., 2019).

All these previous results indicate that learning a better seed dictionary is a crucial step to improve unsupervised cross-lingual word embedding induction and reduce the gap between unsupervised methods and supervised methods, and that GAN-based methods hold the most promise to achieve this goal. The results also indicate that a solution that can handle the full complexity of induction of cross-lingual word embeddings will show improvements in both close and distant languages.

In this paper, we focus on improving the initial step of distribution matching, using GANs (Hartmann et al., 2019). Because the isomorphism assumption is not observed in reality, we argue that a successful GAN-based model must not learn only one single linear mapping for the entire distribution, but must be able to identify mapping subspaces and learn multiple mappings. We propose a multi-adversarial learning method which learns different linear maps for different subspaces of word embeddings.

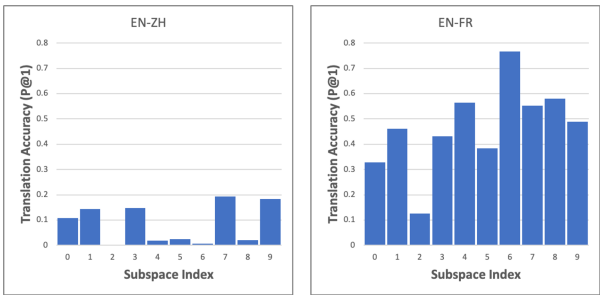

Figure 1: Translation accuracy from English to Chinese and to French for different English subspaces. We only include the top fifty-thousand most frequent English words in the pretrained fastText embeddings. The gold translations comes from Google Translate.

2 Limitations of single-linear mappings

If the assumption by Mikolov et al. (2013) that similar words across source and target languages are related by a single linear relation holds exactly or even approximately , the distance between source and target embedding spaces should be (nearly) evenly minimized during the training of the initial mapping. More specifically, each source subspace should be mapped (nearly) equally well to its corresponding target space, so that the translation ability of the single linear mapping should be similar across different source subspaces.

To verify this expectation, we use the GAN-based system MUSE[1] to train two linear mappings (without refinement) (Conneau et al., 2018). One mapping relates two typologically distant languages, English and Chinese, and the other maps the English space to the space of French - a typologically similar language. We use pretrained FastText embeddings.[2] We split the English space into ten subspaces by running k-means clustering. We evaluate the trained linear mappings by calculating the translation accuracy with precision at one (P@1) —how often the highest ranked translation is the correct one— for each subspace, using the translations from Google Translate as the gold dataset.

[1]https://github.com/facebookresearch/MUSE
[2]https://fasttext.cc/docs/en/pretrained-vectors.html

To reduce the influence of infrequent words, we only consider the first fifty-thousand most frequent source words.

As we can see in Figure 1, the distribution of accuracies of different subspaces is not uniform or even nearly so. This is true for both language pairs, but particularly for the distant languages, where the general mapping does not work at all in some subspaces. Similar phenomena were also discovered by Nakashole (2018) where source words are grouped into different categories. This lack of uniformity in results corroborates the appropriateness of designing a model that learns different linear mappings for different subspaces instead of only learning a single linear mapping for the entire source space.

3 Multi-adversarial CLWE learning

To learn different mappings for different source subspaces, we propose a method for training one GAN for each source subspace. These multi-discriminator GANs encourage the distribution of mapped word embeddings from a specific source subspace to match the distribution of word embeddings from the corresponding target subspace.

The first step of our proposed method is to train a single linear mapping, as in previous approaches. This is used to find aligned subspaces. Our proposed multi-discriminator GAN model then learns the multi-linear mapping. This section starts with the two GAN models, followed by the subspace alignment method, and then describes methods used to improve the GAN training.

3.1 Unsupervised CLWE learning

We first define the task of learning CLWEs and the role of GANs in the previous work of Conneau et al. (2018). Let two monolingual word embeddings $V_s^j = \{v_{s_1}, ..., v_{s_j}\}$ and $V_t^k = \{v_{t_1}, ..., v_{t_k}\}$ be given. In previous work, mapping V_s^j to V_t^k means seeking a linear transformation matrix W, so that the projected vector Wv_i of a source word is close to the vector of its translation in the target language. The basic idea underlying supervised methods is using a seed dictionary of n word pairs $\{(w_{s_1}, w_{t_1}), ..., (w_{s_n}, w_{t_n})\}$ to learn the matrix W by minimizing the distance in (1), where v_{s_i} and v_{t_i} represent the embeddings of w_{s_i} and w_{t_i}. The trained matrix W can then be used to map the source word embeddings to the target space.

$$\min_W \sum_{i=1}^{n} \|Wv_{s_i} - v_{t_i}\|^2 \quad (1)$$

In an unsupervised setting, the seed dictionary is not provided. Conneau et al. (2018) propose a two-step system where the seed dictionary is learned in an unsupervised fashion. In a first step, they use GANs to learn an initial linear transformation matrix W and use this to induce a seed dictionary by finding the translations of the first ten-thousand most frequent source words. In a second step, the seed dictionary is used to refine the initial matrix W. In this work we focus on the GAN component of this model.

3.2 GAN learning of a single linear mapping

Previous GAN-based systems learn a single linear mapping from the source embedding space to the target embedding space. In such models, a source word is trained against a target word sampled from the whole target distribution, and the resulting single linear mapping is applied to all the source words. We first introduce the basic GAN architecture for CLWE of Conneau et al. (2018). We use this model as our comparative baseline and as the initial stage of our proposed method.

A standard GAN model plays a min-max game between a generator G and a discriminator D (Goodfellow et al., 2014). The generator learns from the distribution of source data and tries to fool the discriminator by generating new samples which are similar to the target data.

When we adapt the basic GAN model to learning CLWE, the goal of the generator is to learn the linear mapping matrix W. The discriminator D detects whether the input is from the distribution of target embeddings p_{v_t}. Conneau et al. (2018) use the loss functions in (2) and (3) to update the discriminator and the generator, respectively. $G(v_s) = Wv_s$, and $D(v)$ denotes the probability that the input vector v came from the target distribution p_{v_t} rather than the generator applied to samples from the source distribution p_{v_s}.

$$l_D = -\log D(v_t) - \log(1 - D(G(v_s))) \quad (2)$$

$$l_G = -\log D(G(v_s)) - \log(1 - D(v_t)) \quad (3)$$

The parameters of both generator and discriminator are updated alternatively by using stochastic gradient descent. However, a number of additional methods are needed for robust reliable training of such GANs, which are discussed in Section 3.5.

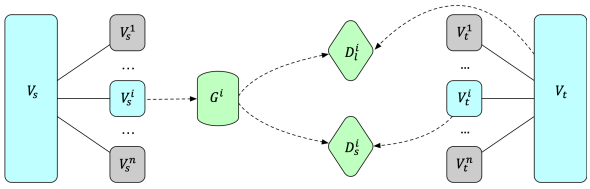

Figure 2: Architecture of our multi-discriminator model. The generator G^i for each source subspace V_s^i is trained against the discriminator D_s^i for the aligned subspace V_t^i and a whole-language discriminator D_l^i.

3.3 GAN learning of a multi-linear mapping

Unlike previous work, we propose learning different linear mappings for different source subspaces. We propose a multi-discriminator GAN where a source word from one subspace is trained against a target word sampled from the aligned target subspace.

For each subspace of source embeddings, we propose a multi-discriminator adversarial model to train the specific mapping for vectors that belong to this subspace. As the architecture in Figure 2 illustrates, the generator of the given source subspace i takes the vector sampled from the sub-distribution as input and maps it to the target language. Differently from standard GANs, the mapped vector $G^i(v_s^i)$=$W^i v_s^i$ will be fed into two discriminators. First, a subspace-specific discriminator D_s^i judges whether the vector has come from the correspondent target subspace i. Thus, we use the vectors sampled from both source and target subspaces to train D_s^i. Second, a normal language discriminator D_l^i judges whether the vector has come from the whole target distribution. This language discriminator helps avoid local optima for the specific subspace.

Both discriminators are two-layer perceptron classifiers. Except for the different sampling ranges, their loss function is similar to equations (2) and (3):

$$l_{D_l^i} = -\log D(v_t') - \log(1 - D(G(v_s'))) \quad (4)$$

$$l_{D_s^i} = -\log D(v_t^i) - \log(1 - D(G(v_s^i))) \quad (5)$$

where v_s' and v_t' are sampled from the 75-thousand most frequent source and target words,[3] and v_s^i and v_t^i are sampled from the specific source subspace V_s^i and its corresponding target subspace V_t^i.

[3]We use different language discriminator models D_l^i for each subspace i, even though their training samples all come from the same distributions. This leads to more stable training, presumably because initially these language discriminators are randomly different.

Since the outputs of both discriminators are used for training the generator, the loss function of the subspace-specific generator G^i can be written as:

$$l_{G^i} = -\lambda(\log D_l^i(G^i(v_s^i)) + \log(1 - D_l^i(v_t'))) - (1-\lambda)(\log D_s^i(G^i(v_s^i)) + \log(1 - D_s^i(v_t^i))) \quad (6)$$

where λ is a coefficient that we call global confidence, which balances the contributions of the two discriminators in updating the generator. In practice, we find that setting λ to 0.5 for each subspace works well for the final result.

Additionally, as the similarities between the entire distribution and the distribution of different subspaces are different, it is justified to use different lambdas for different subspaces instead of using a single one. We therefore propose a metric to set λ dynamically, based on the proportion of the eigenvalue divergence between the two subspaces and the eigenvalue divergence between the whole source and target distributions, as shown in (7). In this paper, we only report results with dynamic λ.

$$\lambda = \frac{EVD(V_s^i, V_t^i)}{EVD(V_s, V_t)} \quad (7)$$

The eigenvalue divergence between two embedding distributions V_1 and V_2 can be computed as shown in (8), where $e_k^{V_1}$ and $e_k^{V_2}$ represent the eigenvalues of V_1 and V_2.

$$EVD(V_1, V_2) = \sum_{k=1}^{d} (\log e_k^{V_1} - \log e_k^{V_2})^2 \quad (8)$$

All subspace-specific generators are initialized with the single linear mapping discussed in Section 3.2.

3.4 Subspace alignment

The above multi-discriminator GAN assumes that we have an alignment between source subspaces and target subspaces. We first present the method we use to produce aligned subspaces in both source and target distributions, and then the clustering method we use to find coherent subspaces, which are both important for the model's improved performance.

If we want to encourage words from a specific source subspace to be trained against words from a matching target subspace, we need to align the two cross-language subspaces. The second problem we need to solve for our multi-adversarial method

to work is how to discover this alignment. Although metrics such as Gromov-Hausdorff distance (GH) (Patra et al., 2019) and Eigenvalue Divergence (EVD) (Dubossarsky et al., 2020) can be used to measure the similarity between two distributions and find the most similar target subspace for a given source subspace, matching between two sub-distributions may amplify any bias generated during the clustering.

To avoid this problem, we only run the clustering on the source side. For a given source embedding space V_s, we denote its subspaces after clustering as $\{V_s^1, V_s^2, ..., V_s^i, ..., V_s^n\}$, where n represent the number of subspaces. To align target words to their matching source subspace, we propose to first learn a single linear mapping from source to target space using the GAN-based method (without refinement) described previously, and then use the transpose of this linear mapping to retrieve the translation of each target word in the source language (using cross-domain similarity local scaling, defined below in Section 3.5). The subspace index of the target word is then set to the subspace index of this translation. In this way, the target embedding space V_t is partitioned into as many subspaces as the source embedding space, denoted as $\{V_t^1, V_t^2, ..., V_t^i, ..., V_t^n\}$. This gives us aligned subspace pairs (V_s^i, V_t^i).

Although the single linear mapping from source language to target language is not good enough to get accurate translations, our experiments indicate that it is a good method to produce a subspace alignment. A possible reason for this result is that the clustering on the source language has already grouped similar words. Therefore, even if a translation turns out to be incorrect, it usually has the same subspace index as the best translation.

Parameter-free hierarchical clustering A major issue in clustering an embedding space is how to find a clustering that adapts to the space, without fixed parameters. To avoid having to identify the number of subspaces in advance, we use hierarchical clustering. Recent work proposes a parameter-free method called First Integer Neighbor Clustering Hierarchy (FINCH) (Sarfraz et al., 2019), which we use in this paper. Traditionally, clustering methods split a given space of vectors into different clusters by calculating the distances between the centroid and the other vectors. FINCH is developed based on the observation that the first neighbour of each vector is a sufficient statistic to find links in the space, so that computing the distance matrix between all the vectors is not needed (Sarfraz et al., 2019). For a given vector space, one first computes an adjacency link matrix using the equation in (9).

$$A(i,j) = \begin{cases} 1 & if\ j = \kappa_i^1\ or\ \kappa_j^1 = i\ or\ \kappa_i^1 = \kappa_j^1 \\ 0 & otherwise \end{cases} \tag{9}$$

where i, j denote the indices of vectors and κ_i^1 represents the index of the first neighbour of the vector with index i. The connected components can then be detected from the adjacency matrix A by building a directed or undirected graph on A. No parameter needs to be set. When the clustering on the original first level (original data) is completed, the centroid of each cluster can then be considered as a data vector for the next level and a new level of clustering is computed using the same procedure. In theory, all the vectors will eventually be gathered into a single cluster. In practice, we find that using the clusters of the last level or the second-to-last level works well for our system.[4]

3.5 Training the GANs

Training the GANs described in Sections 3.2 and 3.3 can be challenging. Based on previous work and our experience, we employ the following techniques during training.

Orthogonalization Previous work shows that enforcing the mapping matrix W to be orthogonal during the training can improve the performance (Smith et al., 2017). In the system of Conneau et al. (2018), they follow the work of Cisse et al. (2017) and approximate setting W to an orthogonal matrix with $W \leftarrow (1+\beta)W - \beta(WW^\top)W$. This orthogonalization usually performs well when setting β to 0.001 (Conneau et al., 2018; Wang et al., 2019).

Cross-Domain Similarity Local Scaling The trained mapping matrix W can be used for retrieving the translation for a given source word w_s by searching a target word w_t whose embedding vector v_t is close to Wv_s. But Conneau et al. (2018) showed that using cross-domain similarity local scaling (CSLS) to retrieve translations is more accurate than standard nearest neighbor techniques and can reduce the impact of the hubs problem

[4] In the code of Sarfraz et al. (2019), the last level means the level before grouping all the data vectors into a single cluster.

(Radovanović et al., 2010; Dinu et al., 2015). Instead of just considering the distance between Wv_s and v_t, CSLS also takes into account the neighbours of v_t in the source language by minimising $(2\cos(Wv_s, v_t) - r_t(Wv_s) - r_s(v_t))$, where $r_t(Wv_s)$ denotes the mean similarity between a Wv_s and its neighbours in the target language, while $r_s(v_t)$ represents the mean similarity between v_t and its neighbours in the source language.

Model selection criterion The cosine-based model selection criterion is another important component of adversarial training for selecting the best mapping matrix W. More specifically, at the end of each training epoch, the current mapping is used to translate the ten-thousand most frequent source words into target words and calculate the average cosine similarity between the source vectors and the target vectors. This cosine-based criterion has been shown to correlate well with the quality of W (Conneau et al., 2018; Hartmann et al., 2019).

Random restarts Previous work (Vulić et al., 2019; Glavaš et al., 2019) shows that using GANs to train the mapping matrix W is not stable. Hartmann et al. (2019) propose to solve this problem with the random restart technique. More specifically, before going to the step of refinement, they randomly train ten mapping matrices, choosing only the best model among them for the next step. The best model is selected with the unsupervised model selection criterion. Their experiments show that this model selection method has the best performance on bilingual lexicon induction. We follow (Vulić et al., 2019; Glavaš et al., 2019) and apply the same random restart technique to train the single linear mapping and use it to initialize each subspace-specific generator.

4 CLWE mapping refinement

As in previous work, after GANs have been used to find a mapping from source to target word embeddings, a refinement step can be used to improve this mapping. Refinement involves first inducing a seed dictionary of word translations, and then refining the mapping using this seed dictionary.

Bidirectional seed dictionary induction Using the mapping learned with adversarial training, the translations $(w_{t_1}, w_{t_2}, ..., w_{t_{10000}})$ for the top ten-thousand source words $(w_{s_1}, w_{s_2}, ..., w_{s_{10000}})$ are retrieved and then back-translated into the source language $(w'_{s_1}, w'_{s_2}, ..., w'_{s_{10000}})$. The mutual translation pairs (w_{s_i}, w_{t_i}) such that $w_{s_i} = w'_{s_i}$ constitute the seed dictionary. This guarantees that the induced seed dictionary will be bidirectional.

Mapping refinement The refinement step is based on the Procrustes Analysis (Schönemann, 1966). With the seed dictionary, the mapping can be updated using the objective in equation (1), and forced to be orthogonal using singular value decomposition (SVD) (Xing et al., 2015). Later work combines the Procrustes Analysis with stochastic dictionary induction (Artetxe et al., 2018) and greatly improves the performance of the standard refinement (Hartmann et al., 2019). More specifically, in order to prevent local optima, after each iteration some elements of the similarity matrix are randomly dropped, so that the similarity distributions of words change randomly and the new seed dictionary for the next iteration varies.

Global and local refinement Refinement can be applied to our multi-linear mapping in two different ways. First, after the training of all the subspace alignments, we can refine a linear relationship between the transformed source embeddings and the target embeddings, like previous unsupervised methods. This we call *global refinement*. It is noteworthy that the combination of the multi-linear mapping trained by our multi-discriminator model and the refined single linear mapping is still multi-linear. Second, we can also refine the mapping of each subspace separately. More concretely, for a given subspace (V_s^i, V_t^i), we build a local seed dictionary and use the local seed dictionary to update the mapping $G^i(v_s^i)$. We call this *local refinement*. We evaluate both global and local refinement in the next section.

5 Experiments

Bilingual lexicon induction (BLI) has become a standard task for evaluating CLWE models. However, according to Glavaš et al. (2019) and Zhang et al. (2020), BLI performance of a given CLWE model doesn't always correlate with performance in other cross-lingual downstream tasks. In this section, we evaluate our proposal on both the task of BLI and the task of cross-lingual document classification (CLDC).

We evaluate our system both with and without refinement. Since GAN-based methods of learning CLWE are often criticized for their instability at

inducing the seed dictionary, we report the average over 10 runs for the BLI without-refinement setting. We include the random restart technique for other tasks and report the result of the best model selected by the unsupervised model selection criterion. We evaluate our model both with global refinement (G-Ref) and local refinement (L-Ref).

BLI setting We use the dataset provided by Conneau et al. (2018) for the task of BLI. This dataset contains high quality dictionaries for more than 150 language pairs. For each language pair, it provides a training dictionary of 5000 words and a test dictionary of 1500 words. This dataset allows us to have a better understanding of the performance of our proposal on many different language pairs. For each language pair, we retrieve the best translations of source words in the test dictionary using CSLS, and we report the accuracy with precision at one (P@1).

CLDC setting We use the multilingual classification benchmark (MLDoc) provided by Schwenk and Li (2018) for the task of CLDC. MLDoc contains training and test documents with balanced class priors for eight languages: German (de), English (en), Spanish (es), French (fr), Italian (it), Japanese (ja), Russian (ru) and Chinese (zh). We follow previous works (Glavaš et al., 2019; Zhang et al., 2020) and train a CNN classifier on English using 10,000 documents and test the classifier on the other seven languages.[5] Each language contains 4000 test documents. The input of the classifier comes from the CLWE models. We report the average accuracy over ten runs.

Language pairs In this paper, we focus on projecting foreign language embeddings into the English space. We choose the eight languages included in MLDoc for both the BLI and CLDC tasks. Within the seven non-English languages, Japanese, Russian and Chinese are languages distant from English and the others are languages similar to English. For the task of BLI, we also investigate Turkish, another language distant from English.

Monolingual word embeddings We use the pretrained FastText embedding models (Bojanowski et al., 2017) for our experiments. These embeddings of 300 dimensions are pretrained on Wikipedia dumps and publicly available.[6] Following previous works, we use the first 200,000 most frequent words for each monolingual embedding model.[7] We apply iterative normalization (Zhang et al., 2019) on each embedding model before training.

BLI Task - with refinement								
	de	es	fr	it	ja	ru	tr	zh
PROC	73.1	83.6	82.2	77.5	37.9	64.3	63.1	40.0
RCSLS	73.1	83.1	83.1	78.9	39.3	64.6	63.1	43.0
MUSE	73.7	83.0	82.2	78.5	29.3	62.7	60.5	38.1
VecMap	73.6	**83.7**	**82.9**	78.5	**34.7**	63.1	**61.3**	36.4
Ours GRef	**74.1**	**83.7**	82.4	**78.6**	34.1	**64.0**	61.2	**38.2**
Ours LRef	66.6	79.3	77.8	70.3	23.7	46.5	39.7	29.7

Table 1: BLI task results with refinement. Bold shows the best score within unsupervised systems and underline shows the best score over all the systems.

BLI Task - without refinement								
	de	es	fr	it	ja	ru	tr	zh
Successful runs averages								
MUSE	53.9	68.9	66.9	**60.7**	14.7	**38.1**	22.3	16.2
VecMap	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ours	**55.5**	**69.3**	**67.3**	59.3	**18.3**	**38.1**	**28.4**	**19.1**
Failures								
MUSE	3	1	0	1	5	5	4	9
VecMap	10	10	10	10	10	10	10	10
Ours	1	1	0	0	5	4	4	8

Table 2: BLI task results for unsupervised models without refinement. We consider accuracy below 2% as failure and report the average accuracy with P@1 over the successful runs. Bold represents the best score.

Baselines The objective of our proposal is to improve the mapping ability of GANs by learning a multi-linear mapping instead of only a single-linear mapping. Therefore, we use the GAN-based system MUSE (Conneau et al., 2018)[8] as our main unsupervised baseline. Since the unsupervised method proposed by Artetxe et al. (2018)[9] is considered a robust CLWE system, we also use it as our second unsupervised baseline (VecMap in the tables). In the setting with refinement, we use the iterative refinement with stochastic dictionary induction for all the unsupervised systems.[10] We also include two supervised systems, Procrustes (PROC) (Conneau et al., 2018) and Relaxed CSLS (RCSLS) (Joulin et al., 2018), to better understand

[5] https://github.com/zhangmozhi/retrofit_clwe

[6] https://fasttext.cc/docs/en/pretrained-vectors.html

[7] The original pretrained Latvian fastText model only consists of 171,000 words.

[8] https://github.com/facebookresearch/MUSE

[9] https://github.com/artetxem/vecmap

[10] We disabled the re-weighting technique since it's not applicable for L-Ref. However, adding re-weighting to VecMap, MUSE and G-Ref doesn't change the gaps between them.

our method. Both PROC and RCSLS are robust supervised systems for learning CLWE and have been widely used previously (Glavaš et al., 2019; Zhang et al., 2020). We also wanted to include the supervised system proposed by Nakashole (2018), which learns multiple local mappings between embedding spaces, but their code is not publicly available.

5.1 Performance on BLI

We report the results of BLI both with and without refinement in Tables 1 and 2, respectively.

The results in Table 2 show a clear improvement from our multi-linear mapping, compared with single-linear GANs. We perform better than MUSE for both average accuracy and number of failures in almost every language pair. The advantages are more striking on distant language pairs.

VecMap is considered the most robust unsupervised model for learning CLWE. However, according to Hartmann et al. (2019), the advantage of VecMap mostly comes from its refinement technique. From the results in Table 1, we can see that when using the same refinement technique, our best model selected from ten random restarts using the unsupervised metric performs as well as VecMap or even better. Our model achieves higher scores on four language pairs and comes close for the other language pairs.

The results shown in Table 1 demonstrate that our model is comparable with supervised systems when using iterative refinement with stochastic dictionary induction and random restarts. We even perform better than PROC and RCSLS on similar language paris such as German and Spanish to English (de-en and es-en).

From the results in Table 1, we can easily see that the global refinement outperforms local refinement. Using local refinement we even perform much worse than our GAN-based baseline. This phenomenon does not surprise us since local refinement can easily lead to overfitting on a given subspace, and we leave the investigation of alternative refinement methods to future work.

5.2 Performance on CLDC

We report the results on the task of CLDC without and with refinement in Tables 3 and 4, respectively.

From the results shown in Table 3, we can see that our multi-linear model continues to maintain its advantage over the single-linear GAN, MUSE, in the setting without refinement. When refinement is added, MUSE becomes a little better than our model on German and Spanish. However, we still perform better on all those languages that are distant from English.

CLDC Task - without refinement							
	de	es	fr	it	ja	ru	zh
MUSE	79.2	**69.7**	71.5	56.4	27.6	56.3	60.8
VecMap	21.4	24.6	23.1	23.7	21.2	23.6	22.3
Ours	**80.1**	67.4	**73.2**	**62.3**	**30.3**	**61.5**	**69.5**

Table 3: CLDC results on MLDoc dataset (Schwenk and Li, 2018) without refinement. Bold represents the best score.

CLDC Task - with refinement							
	de	es	fr	it	ja	ru	zh
PROC	81.4	69.6	70.7	62.9	30.0	64.9	32.0
RCSLS	81.6	70.5	71.1	62.4	29.7	64.3	31.8
MUSE	80.9	69.4	70.6	59.9	28.9	61.1	45.6
VecMap	**81.8**	69.8	**71.0**	**64.0**	28.4	63.2	35.4
Ours G-Ref	80.7	69.2	70.9	62.6	**30.2**	61.3	55.9
Ours L-Ref	79.5	69.1	69.9	62.6	30.1	62.9	**59.9**

Table 4: CLDC results on MLDoc dataset (Schwenk and Li, 2018) with refinement. Bold shows the best score within unsupervised systems and underline shows the best score over all the systems.

Differently from the task of BLI, there is no obvious advantage in the supervised baselines over our multi-linear model both with and without refinement. Conversely, as the results in Table 3 indicate, our model without refinement performs comparably or better than either our supervised baselines or VecMap in the setting with refinement. For example, our model achieves 69.5 of accuracy on Chinese test data, while the best supervised model, PROC, only has 32.0 accuracy.

While CLWE refinement is a necessary step for the BLI task, for the CLDC task our model does not seem to need refinement. As the performance gap illustrated in Figure 3 shows, our model performs worse when adding refinement for languages such as French, Japanese, Russian and Chinese, which includes all the languages distant from English. Furthermore, even for languages where we benefit from refinement, the improvement is limited.

6 Conclusion

In this paper, we propose a multi-adversarial learning method for cross-lingual word embeddings.

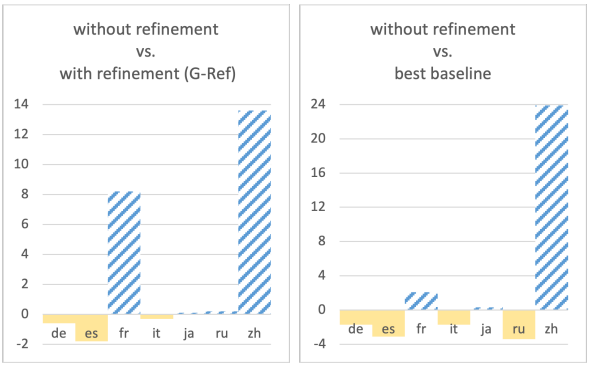

Figure 3: Performance gap on the CLDC task. The left panel represents the gap between our multi-linear model with refinement and without refinement. The right panel represents the performance gap between our model without refinement and the best baseline (best results selected from our supervised and unsupervised baselines). Blue bars indicate the cases where the model without refinement performs better than its competitors. Yellow bars represent the opposite cases.

Our system learns different linear mappings for different source subspaces instead of just learning a single one for the whole source space. The results of our experiments on bilingual lexicon induction and cross-lingual document classification on both close languages and distant languages prove that learning cross-lingual word embeddings with a multi-linear mapping improves performance over a single-linear mapping. Future work will focus on learning multi-linear mappings for contextualized embeddings.

References

David Alvarez-Melis and Tommi S. Jaakkola. 2018. Gromov-Wasserstein Alignment of Word Embedding Spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, volume 1881-1890, Brussels, Belgium.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 451–462, Vancouver, Canada.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume Long Papers, pages 789–798, Melbourne, Australia.

Antonio Valerio Miceli Barone. 2016. Towards Cross-Lingual Distributed Representations without Parallel Text Trained with Adversarial Autoencoders. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 121–126, Berlin, Germany.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. 2017. Parseval Networks: Improving Robustness to Adversarial Examples. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 854–863, Sydney, Australia.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word Translation Without Parallel Data. In *Proceedings of the 6th International Conference on Learning Representations*, pages 1–14, Vancouver, Canada.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving Zero-Shot Learning by Mitigating the Hubness Problem. In *Proceedings of the 3rd International Conference on Learning Representations*, volume Workshop Track, pages 1–10, Toulon, France.

Haim Dubossarsky, Ivan Vulić, Roi Reichart, and Anna Korhonen. 2020. Lost in Embedding Space: Explaining Cross-Lingual Task Performance with Eigenvalue Divergence. *arXiv*, pages 1–10.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (Properly) Evaluate Cross-Lingual Word Embeddings: On Strong Baselines, Comparative Analyses, and Some Misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721, Florence, Italy.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, page 2672–2680, Montréal, Canada.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual Dependency Parsing Based on Distributed Representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1234–1244, Beijing, China.

Mareike Hartmann, Yova Kementchedjhieva, and Anders Søgaard. 2019. Comparing Unsupervised Word Translation Methods Step by Step. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pages 6033–6043, Vancouver, Canada.

Yedid Hoshen and Lior Wolf. 2018. An Iterative Closest Point Method for Unsupervised Word Translation. *ArXiv*.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Herve Jegou, and Edouard Grave. 2018. Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing Crosslingual Distributed Representations of Words. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1459–1473, Mumbai, India.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 1st International Conference on Learning Representations*, pages 1–12, Arizona, USA.

Ndapa Nakashole. 2018. NORMA: Neighborhood Sensitive Maps for Multilingual Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 512–522, Brussels, Belgium.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig. 2019. Bilingual Lexicon Induction with Semi-supervisionin Non-Isometric Embedding Spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistic*, pages 184–193, Florence, Italy.

Milos Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010. Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data. *Journal of Machine Learning Research*, 11:2487–2531.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A Survey Of Cross-lingual Word Embedding Models. *Journal of Artificial Intelligence Research*, 65:569–631.

M. Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. 2019. Efficient Parameter-free Clustering Using First Neighbor Relations. In *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, Long Beach, California, USA.

Holger Schwenk and Xian Li. 2018. A Corpus for Multilingual Document Classification in Eight Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan.

Peter H. Schönemann. 1966. A generalized solution of the Orthogonal Procrustes problem. *Psychometrika*, 31:1–10.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline Bilingualword Vectors, Orthogonal Transformations and the Inverted Softmax. In *Proceedings of the 15th International Conference on Learning Representations*, pages 1–10, Toulon, France.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the Limitations of Unsupervised Bilingual Dictionary Induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume Long Papers, pages 778–788, Melbourne, Australia.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. Do We Really Need Fully Unsupervised Cross-Lingual Embeddings? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 4407–4418, Hong Kong, China.

Ivan Vulić, Sebastian Ruder, and Anders Søgaard. 2020. Are All Good Word Vector Spaces Isomorphic? *arXiv*, pages 1–11.

Haozhou Wang, James Henderson, and Paola Merlo. 2019. Weakly-Supervised Concept-based Adversarial Learning for Cross-lingual Word Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 4419–4430, Hong Kong, China.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1006–1011, Denver, Colorado.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial Training for Unsupervised Bilingual Lexicon Induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1959–1970, Vancouver, Canada.

Mozhi Zhang, Yoshinari Fujinuma, Michael J Paul, and Jordan Boyd-Graber. 2020. Why Overfitting Isn't Always Bad: Retrofitting Cross-Lingual Word Embeddings to Dictionaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Online.

Mozhi Zhang, Keyulu Xu, Ken-ichi Kawarabayashi, Stefanie Jegelka, and Jordan Boyd-Graber. 2019. Are Girls Neko or Shōjo? Cross-Lingual Alignment of Non-Isomorphic Embeddings with Iterative Normalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3180–3189.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA.