Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5555–5562
Marseille, 11–16 May 2020

A Closer Look on Unsupervised Cross-lingual Word Embeddings Mapping

Kamil Pluciński^{1,2}, **Mateusz Lango**^{1,2}, **Michał Zimniewicz**^{1}
^{1} Poznan Supercomputing and Networking Center, Poznań, Poland
^{2} Institute of Computer Science, Poznan University of Technology, Poznań, Poland
kamil.plucinski97@gmail.com, mlango@cs.put.poznan.pl, mzimniewicz@man.poznan.pl

Abstract

In this work, we study the unsupervised cross-lingual word embeddings mapping method presented by Artetxe et al. (2018). First, we successfully reproduced the experiments performed in the original work, finding only minor differences. Furthermore, we verified the method's robustness on different embedding representations and new language pairs, particularly these involving Slavic languages like Polish or Czech. We also performed an experimental analysis of the impact of the method's parameters on the final result. Finally, we looked for an alternative way of initialization, which directly relies on the isometric assumption. Our work confirms the results presented earlier, at the same time pointing at interesting problems occurring while using the method with different types of embeddings or on less-common language pairs.

Keywords: word embeddings, unsupervised word embedding mapping, cross-lingual mapping, bilingual lexicon construction

1. Introduction

Adapting current NLP technologies to a larger set of languages is an important challenge for both industry and academia. One of the common barriers in applying machine learning approaches to new languages is the lack of proper datasets. Fortunately, this issue can be sometimes circumvented through transfer learning techniques. For instance, such techniques can enable a system trained on dataset for one language to perform well on the same task for another language (Ni et al., 2017). Such modern transfer techniques frequently leverage word embeddings as a medium for knowledge transfer.

A word embedding is a representation of a word as a multidimensional vector whose position in the space reflects semantic and/or syntactic similarities to other words in the same language. The majority of methods for the formation of word embeddings require only a single, unannotated corpus, which fosters their usage on a wide range of languages (Grave et al., 2018). Although word embeddings are available for many languages, their availability does not completely solve the problem of cross-lingual transfer. The embeddings for different languages are trained separately on distinct corpora, so they do not share the same vector space. This means that words representations with similar meaning in distinct languages can be very different. Not surprisingly, methods aligning word embeddings from different languages have been proposed.

One of the first such approaches was proposed by Mikolov et al. (2013b) and it used stochastic gradient descent to learn a translation matrix which mapped vectors from one space to another i.e. it mapped words from one language to another. The method was supervised and required a dataset of about five thousand pairs of translated words to learn the mapping. This still could be a problem for some languages.

Later, it was noticed that learning a cross-lingual mapping is also possible with very small supervision. For instance, Artetxe et al. (2017) proposed a method which uses a dictionary of only 25 words or numbers. Other related approaches with weak supervision rely on words with identical spelling in both languages (Smith et al., 2017; Søgaard et al., 2018a). More recently, several methods for fully unsupervised bilingual lexicon induction were proposed, including these relying on adversarial learning (Conneau et al., 2017a; Zhang et al., 2017) and on 3D point cloud matching (Hoshen and Wolf, 2018). Notably, Artetxe et al. (2018) proposed a fully unsupervised method that provides state-of-the-art results for bilingual mapping, obtaining better results than supervised algorithms. The method consists of an iterative self-learning procedure with a specialized dictionary initialization technique. For a detailed review of cross-lingual word embeddings methods, please refer to (Søgaard et al., 2019).

In this work, we reproduce and verify the results of the method presented by Artetxe et al. (2018) and take a closer look on its properties. First, we test word embedding mapping on two new Slavic languages - namely Czech and Polish. We investigate the impact of method's parameters values on the final accuracy. Parameters such as the size of initialization dictionary or the threshold for early stopping are taken into account. Studying sensitivity of method's parameters is particularly important in the unsupervised setting since a validation set cannot be used to tune them. Moreover, we analyze the method's performance on other word embedding representation, namely fastText (Bojanowski et al., 2016), assessing its accuracy on other embeddings then only those constructed by word2vec (Mikolov et al., 2013a). Finally, we explore a new method for unsupervised initialization, which leverages the same hypothesis as the original method, raising questions about the reasons of the method's good performance.

The rest of the paper is organized as follows. In Section 2. we shortly describe the method of Artetxe et al. (2018) which results are reproduced in this work. The following section contains a description of the reproduced experiments as well as a discussion of the obtained results. In Section 4., we further investigate the properties of the cross-lingual mapping method by verifying its robustness on new language pairs and word embedding representations. The impact of method's parameters and a new unsupervised initialization strategy are also examined there. In the last section, we summarize our work and draw lines of further research.

2. Reproduced Paper

The unsupervised cross-lingual word embedding mapping method (Artetxe et al., 2018), which is studied in this paper, is one of the first unsupervised methods proposed in the literature for cross-lingual word embedding mapping. Moreover, its authors demonstrate that their method requires less computation time and is much more stable, providing high performance not only on closely-related languages.

The method relies on the idea of self-improvement/self-learning. It begins with a noisy mapping of two embedding spaces and determines (also noisy) pairs of translated words based on that mapping. Having translated word pairs, it is possible to calculate a new mapping that better aligns the representations of corresponding words. However, if a better mapping is available, also better translations pairs can be constructed, so the procedure starts from the beginning.

In the following subsections, we briefly describe the self-learning procedure and then show how the initial mapping is constructed.

2.1. Self-learning

Let us introduce a notation. The input to the method consists of two matrices X and Z whose rows contain word embeddings from a given language. Our task is to learn an orthogonal (linear) transformation of these embeddings/matrices to a common space, denoted by W_X and W_Z, respectively. Having the embeddings XW_X and ZW_Z mapped into a common space, one can construct a dictionary of corresponding word pairs from both languages. This dictionary is represented by a matrix D where $D_{ij} = 1$ if X_i represents the translation of Z_j, and $D_{ij} = 0$ otherwise.

As mentioned earlier, the self-learning procedure consists of two steps. In the first step, we take initial W_X and W_Z mappings and construct the dictionary D. Since, after transformation, the vectors in XW_X and ZW_Z should be aligned i.e. similar words from both languages should have similar vector representations, the dictionary of translated word pairs can be simply constructed by nearest neighbor search. More formally, assuming normalized vectors and cosine similarity, $D_{ij} = 1$ if $j = \arg\max_k X_i W_X (Z_k W_Z)^T$, and $D_{ij} = 0$ otherwise.

The result of searching for translations with the nearest neighbor algorithm can be highly influenced by the hubness phenomenon, which occurs in a high-dimensional space. To alleviate this issue, instead of standard cosine similarity, Cross-domain Similarity Local Scaling (Conneau et al., 2017b) is used as the similarity function. CSLS is defined as

$$CSLS(x, z) = 2cos(x, z) - r_T(x) - r_S(z) \qquad (1)$$

where $r_T(x) = \frac{1}{k}\sum_{z \in N_T(x)} cos(x, z)$ and $N_T(x)$ denotes the set of $k = 10$ nearest neighbors of x in the target language space. $r_S(z)$ is defined in the corresponding way. This function reduces the similarity of each word by subtracting its average similarity to the nearest neighbors, punishing the "hub" words.

Once the dictionary is constructed, the method determines the optimal transformation of both spaces to a common one, making the transformed embeddings of translations in D as similar as possible. This is achieved by solving the following optimization problem:

$$\operatorname*{arg\,max}_{W_X, W_Z} \sum_i \sum_j D_{ij}((X_{i*}W_X) \cdot (Z_{j*}W_Z)) \qquad (2)$$

$$s.t.\ W_X \text{ and } W_Z \text{ being orthonormal}$$

which is equivalent to the orthogonal Procrustes problem (Schönemann, 1966). It can be effectively solved by SVD decomposition of $X^T DZ$ matrix into USV^T where $W_X = U$ and $W_Z = V$. Since the mapping transformations were updated, the method returns to the dictionary construction step as it is now possible to find translations more consistent with the current alignment of vector spaces.

2.2. Unsupervised initialization

In the previous subsection, we described an algorithm for learning cross-lingual mapping, which starts from some initial, noisy mapping. It can seem that the procedure can start with any randomly chosen mapping or with a random dictionary, but actually, the algorithm does not converge to an accurate mapping if starting from a completely random solution. Hence, the authors of (Artetxe et al., 2018) propose an unsupervised initialization method that constructs a dictionary with very low accuracy (e.g. 0.52% for English-Italian) but which is better than random, delivering a sufficiently good starting point for the further learning process.

The initialization procedure leverages an assumption about the isometry of the embedding spaces of both languages. More concretely, it assumes that even though matrices X and Z contain embeddings in different spaces, the similarities between words in both languages should be the same, i.e., the similarity matrices $M_X = XX^T$ and $M_Z = ZZ^T$ are equivalents up to the permutation of their columns and rows. In fact, finding such permutation is equivalent to the construction of a translation dictionary. Unfortunately, the search space of all permutations is enormous and grows exponentially with the size of the vocabulary. This issue led to the proposal of a heuristic approach rather than looking for the best solution among all possible permutations.

Algorithm 1 Embedding preprocessing

```
function PREPROCESS(embedding)
    $emb$ ← normalize(embedding)
    $M \leftarrow USU^T$ where $USV^T = SVD(emb)$
    $sorted(M)$ ← sort values in $M`s$ rows
    $result$ ← normalize($sorted(M)$)

function NORMALIZE(embedding)
    $emb$ ← divide each vector by its norm
    make vectors $emb$ zero mean
    $emb$ ← divide each vector by its norm
    $result \leftarrow emb$
```

The unsupervised initialization consists of a specific preprocessing of embedding matrices, presented in Algorithm 1, and the construction of a dictionary by earlier described nearest neighbor search with CSLS. The main idea of the proposed heuristic is that if the isometric assumption holds, corresponding words should have the same values in M_X and M_Z except for their order. Therefore by sorting values in the rows of similarity matrices, one gets the same representations for the corresponding words. Since the isometric assumption does not hold perfectly, the nearest neighbor search is used instead of an exact match.

2.3. Additional techniques

The method consists of several additional techniques which we will not describe in detail. First, during the dictionary construction, it finds the translations $D_{X\to Z}$ by nearest neighbor search from language X to Z, but then the same operation is performed in reverse order, i.e., translations $D_{Z\to X}$ from Z to X are found. The final matrix D is determined as a sum of these two matrices. Moreover, the elements of similarity matrix used by the nearest neighbor search are randomly set to 0. The proportion of zeroing elements p is decreased during the training when no improvement is observed for 50 iterations. Next, to make computations faster, the method uses only 20K most frequent words during training. Finally, as the last step of the method, the symmetric re-weighting of W_X and W_Y matrices is performed. For the full description of these methods, please refer to the original paper.

3. Reproduced Experiments

We reimplemented the method in Python and repeated the experiments presented in the original paper. Our implementation is publicly available on Gitlab[1]. The original paper is written clearly with an accurate description of the method and additional techniques, which enables smooth implementation. Nevertheless, we must admit that the reconstruction of some details would not be possible without the source code, which is made available by the authors online. For instance, they use 4 000 embeddings during the initialization process, 20 000 during the learning process, and 200 000 during the validation process. The first two cardinalities are mentioned in the paper, but the last one is not. Similarly, the fact that the word embeddings should be sorted by word's occurrence frequency is not clearly stated. The majority of the datasets needed for reproducing the experiments can be downloaded automatically by the script provided in the authors' code repository. Nevertheless, the 50-dimensional embeddings and validation sets by (Zhang et al., 2017) are not indicated in the script, and we have downloaded them separately. The datasets used for the evaluation of the method are split into train and test parts. It is not clear on which parts the results were calculated in the original paper. Since the approach is fully unsupervised, the results could be reported on train, test, or even on both parts of a dataset joined together. Basing on the obtained accuracies, we determined that only the test parts of the datasets were used for evaluation. According to our experiments, such evaluation is more challenging because the test parts usually contain a substantially higher number of rare words than the training parts.

Each experiment was repeated ten times, and, following the original paper, we report averaged accuracy over all runs, the best obtained accuracy, the number of successful runs (meaning the runs completed with accuracy higher than 5%), and the computation time in minutes. The results of method's evaluation on datasets from (Zhang et al., 2017) are reported in Table 1. The outcome of our reproduction is compatible with the results reported by the method's authors as differences in accuracy are never larger than 1%. Evident differences are noticeable in reported computation time. Our experiments take about four times longer than those reported in the paper. However, this discrepancy can be explained by the differences in the hardware used in both experiments. During our experiments, we have used a Tesla K80 GPU card, whereas a stronger Titan XP was used in the original experiments. We have also obtained an interesting result by training the method in the opposite direction. For instance, having English to Italian translation task, we trained the method in the Italian to English direction and then used learned mappings to translate English words to Italian. Surprisingly, by this simple change, we achieved increased accuracy of about 3-4% for translations from Romance languages (IT-EN and ES-EN pairs), whereas for Turkish the result was lower by approx. 2%.

Furthermore, we reproduced the method's results on more difficult, 300-dimensional embedding datasets. On these datasets, we performed ablation tests, following the original paper. The results of those experiments are presented in Table 2 and 3, respectively.

Generally, the reproduction of results reported in the original paper was successful, with only two major differences. First, for English-Italian pair the method was able to achieve a pretty good result even without the unsupervised initialization i.e., starting with a completely random solution. Nevertheless, it occurred only once during 10 repetitions and did not question the authors' conclusion that the unsupervised initialization is a key part of their method.

The second difference is a significant discrepancy between reported accuracies of the ablation test for CSLS similarity measure. Artetxe et al. (2018) reports that the method without CSLS obtains 0% accuracy on all languages, indicating it as a critical element of the approach. In our experiments, the method without CSLS admittedly obtained inferior results, but it was always successful in converging to a decent solution for all tested languages. We do not know what is the source of this difference as we used the unmodified cosine similarity instead of CSLS during the ablation test, which we consider a natural choice. It is not clear what similarity measure was used by the authors in their ablation test.

4. A closer look on method's properties

In this section, we describe our experiments, which further investigate the properties of the method. First, we check the influence of the method's parameters on the final results, which is particularly important as in the unsupervised setting, its tuning is not possible. Next, we verify the method's stability on new pairs of languages, evaluating its applicability to Slavic languages. The method was tested originally only on word2vec embeddings, so we also study the impact of changing the embedding algorithm to another method.

[1] `https://gitlab.com/kamil.plucinski97/umt` commit hash: b4ee21f4, tag: v.1.1

	ES-EN				IT-EN				TR-EN			
	best	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**
Original paper	76.43	76.28	10	0.6	66.96	66.92	10	0.9	36.1	35.93	10	1.7
Reproduction	75.55	75.30	10	2.2	67.56	67.48	10	4.2	36.81	36.38	10	5.6
Training in the opposite direction	78.87	78.83	10	1.9	71.37	71.16	10	3.7	34.40	34.08	10	6.1

Table 1: Results reported in the original paper and those obtained by our implementation on the dataset of (Zhang et al., 2017). The method was run ten times: "avg" is the averaged accuracy, "best" is the best-achived accuracy, "s" is the number of successful runs (>5% accuracy), and "t" is the average computation time.

	EN-IT				EN-DE				EN-FI				EN-ES			
	best	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**
Original paper	48.53	48.13	10	8.9	48.47	48.19	10	7.3	33.5	32.63	10	12.9	37.6	37.33	10	9.1
Reproduction	48.47	48.09	10	30.5	48.67	48.34	10	27.0	33.64	32.79	10	39.2	37.93	37.47	10	34.3

Table 2: Results reported in the original paper and those obtained by our implementation on the dataset of (Dinu et al., 2014) with extensions of (Artetxe et al., 2017; Artetxe et al., 2018).

Finally, we call the isometric hypothesis into question and perform experiments where the method is initialized by another technique that heavily relies on that hypothesis.

4.1. Checking the sensitivity of method's parameters

4.1.1. The size of dictionary

In the reproduced work, there is a clear statement that the initialization is a critical element that makes the whole process work. This claim was confirmed in the ablation test, so we decided to verify the impact of the dictionary size used for the initialization. Besides the dictionary of 4000 words, we decided to test the use of smaller dictionary sizes, which would limit both the memory and computational requirements of the approach. Concretely, we tested the sizes on the logarithmic scale, starting with 4000 words and halving the dictionary size until reaching 500. The results of our experiment are presented in Table 4.

Unfortunately, it seems that using smaller dictionaries can negatively impact the method's performance, although for many languages using smaller dictionaries is possible. The largest size of 4000 was the only size for which all the runs can be considered successful (accuracy >5%). However, for some languages, such as German and Italian, initializing method with only 500 pairs results with almost the same accuracy. For Finish, using 1000 pairs seems to be enough, but for Spanish lowering the dictionary size negatively impacts the accuracy.

4.1.2. Minimal improvement threshold

Minimal improvement threshold and the number of iterations without improvement are another method's parameters. Both those parameters are strongly related. The first one controls the difference in the optimized function value that is considered an improvement. The second indicates how many iterations without the goal function improvement are performed before reducing the similarity matrix dropout level or ending the learning process. Both parameters affect the number of iterations performed at a given dropout level. As there is a strong influence of both parameters on the same property, we decided to test only one of them: minimal improvement threshold. We tested three threshold levels: 10^{-6} (original), 10^{-4} and 10^{-2}.

The results presented in Table 5 show fluctuations of accuracy around 0.2% in both directions. Due to the significant reduction in the number of iterations, we observe substantial savings in the execution time of the algorithm. Lowering the threshold to 10^{-2} shortens runtime by more than a half without losing the accuracy of the translation. Therefore, we think that in practice, one can successfully use a less restrictive threshold parameter, which speeds up the method without affecting the translations.

4.2. Studying method's robustness on new languages and word embeddings representations

The authors of the reproduced paper claim that previous methods were evaluated "on favorable conditions, using comparable corpora or closely-related languages" and that they propose a fully unsupervised method that is robust. Although the experiments in the original paper demonstrated that their method is more stable than alternative approaches, we decided to test it in further by extending the experiment with Slavic languages (concretely Polish and Czech) as they were not studied earlier. Moreover, their experiment only considered translation from or to English, which, in our opinion, was a severe limitation of their experimental setup. Hence, we verified the accuracy of the translations between another two languages i.e. Polish and Spanish. In the original paper, the method's performance was tested only on embedding representations obtained by word2vec. In this paper, we study the method's robustness to a different embedding algorithm. We selected fastText embeddings as they are inherently different from word2vec ones since the subword information is used during its construction. They also demonstrated to work better for languages exhibiting grammatical declensions such as German and Russian on different tasks in earlier studies (Bojanowski et al., 2016).

The addition of new languages pairs to the experiment required the construction of new validation datasets. The method for the construction of our validation sets was inspired by the approach presented by Dinu et al. (2014). First, we split English words into five groups basing on its frequency. Then, from each group we randomly se-

		EN-IT				**EN-DE**				**EN-FI**				**EN-ES**			
		best	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**
Results	**Original**	48.53	48.13	10	8.9	48.47	48.19	10	7.3	33.5	32.63	10	12.9	37.6	37.33	10	9.1
	Reprod.	48.47	48.09	10	30.5	48.67	48.34	10	27.0	33.64	32.79	10	39.2	37.93	37.47	10	34.3
Unsup. init.	**Original**	0.07	0.02	0	16.5	0	0	0	17.3	0.07	0.01	0	13.8	0.13	0.02	0	15.9
	Reprod.	**47.87**	**4.82**	**1**	**61.4**	0.20	0.05	0	61.0	0.07	0.01	0	58.3	0.07	0.01	0	55.3
Stochastic	**Original**	48.2	48.2	10	2.7	48.13	48.13	10	2.5	0.28	0.28	0	4.3	37.8	37.8	10	2.6
	Reprod.	48.47	48.47	10	6.8	48.07	48.07	10	5.6	0.14	0.14	0	16.5	37.93	37.93	10	6.6
Cutoff	**Original**	46.87	46.46	10	114.5	48.27	48.12	10	105.3	31.95	30.78	10	162.5	35.47	34.88	10	185.2
	Reprod.	46.80	46.51	10	559.9	48.40	47.98	10	627.2	30.97	30.33	10	767.8	35.80	35.02	10	1039.9
CSLS	**Original**	0	0	0	15.0	0	0	0	13.8	0	0	0	13.1	0	0	0	14.1
	Reprod.	**43.33**	**42.53**	**10**	**5.7**	**43.20**	**42.78**	**10**	**3.8**	**27.88**	**27.25**	**10**	**8.1**	**33.00**	**32.62**	**10**	**5.7**
Bidirectional	**Original**	46	45.37	10	5.6	48.27	48.03	10	5.5	31.39	24.86	8	7.8	36.2	35.77	10	7.3
	Reprod.	47.40	47.07	10	18.3	48.93	48.55	10	15.3	32.58	31.54	10	22.3	37.20	36.61	10	17.5
Re-weighting	**Original**	46.07	45.61	10	8.4	48.13	47.41	10	7.0	32.94	31.77	10	11.2	36	35.45	10	9.1
	Reprod.	45.87	45.45	10	20.8	47.67	47.31	10	17.4	32.94	31.95	10	26.2	36.20	35.68	10	21.6

Table 3: The results of ablation test performed with our implementation and the results reported in the original paper. The evaluation was performed on the datasets of (Dinu et al., 2014) with extensions of (Artetxe et al., 2017; Artetxe et al., 2018).

	500			**1000**			**2000**			**4000**		
	best	**avg**	**s**	**best**	**avg**	**s**	**best**	**avg**	**s**	**best**	**avg**	**s**
EN-ES	37.60	37.19	10	39.07	26.50	7	37.73	29.94	8	38.07	37.12	10
EN-FI	33.29	16.58	5	34.06	32.98	10	33.78	32.55	10	33.64	32.91	10
EN-DE	48.33	48.11	10	48.40	48.19	10	48.60	48.26	10	48.53	48.36	10
EN-IT	48.40	48.01	10	48.27	47.93	10	48.47	47.89	10	48.53	48.13	10

Table 4: Results with various initialization dictionary sizes on the datasets of (Dinu et al., 2014) with extensions of (Artetxe et al., 2017; Artetxe et al., 2018).

lected 300 words and looked for the most certain translation in OpenSubtitles corpus from OPUS (Tiedemann, 2012). For new languages, we use 300-dimensional embeddings, similarly to those used in earlier experiments with the datasets of (Dinu et al., 2014). The word2vec word embeddings for the Polish language were downloaded from Wikipedia2vec (Yamada et al., 2018) and cleaned by removing embeddings of tokens consisting of multiple words. Such multi-word expressions do not appear in the embeddings of other languages. The fastText embeddings for all languages were downloaded from the official website[2] (Grave et al., 2018). Since fastText embeddings do not always contain embeddings for all the words in the validation tests, we decided to exclude such words from the evaluation so as not to punish the method for the translations that cannot be found. We were unable to find suitable word2vec embeddings for the Czech language, so we report its accuracy only on fastText.

The results of the experiment are presented in Table 6. We have expected a higher quality mapping of fastText embeddings for words with a lower frequency and languages with rich morphology such as German since fastText performs significantly better on the word similarity task for such languages (Bojanowski et al., 2016). Surprisingly, the mapping

[2] These embeddings were trained using CBOW with position-weights using character 5-grams, 300 dimensions, a window of size 5 and 10 negative samples. For every language Common Crawl and Wikipedia corpora were used.

	EN-IT				**EN-DE**				**EN-FI**				**EN-ES**			
threshold	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**
10^{-6}	48.47	48.09	10	30.5	48.67	48.34	10	27.0	33.64	32.79	10	39.2	37.93	37.47	10	34.3
10^{-4}	48.47	48.17	10	20.6	48.47	48.33	10	16.4	33.85	33.24	10	29.2	38.00	37.27	10	22.3
10^{-2}	48.73	48.22	10	13.0	48.60	48.33	10	13.4	33.50	32.51	10	17.1	37.93	37.31	10	13.6

Table 5: The method's performance with different minimal improvement threshold parameter on the datasets of (Dinu et al., 2014) with extensions of (Artetxe et al., 2017; Artetxe et al., 2018).

	EN-IT				EN-DE				EN-FI				EN-ES			
	best	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**
Word2Vec	48.47	48.09	10	30.5	48.67	48.34	10	27.0	33.64	32.79	10	39.2	37.93	37.47	10	34.3
fastText	57.47	57.09	10	45.7	0.36	0.13	0	60.1	0.00	0.00	0	52.4	57.28	56.80	10	44.1

	EN-PL				ES-PL				EN-CS			
	best	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**	**best**	**avg**	**s**	**t**
Word2Vec	30.49	29.92	10	45.7	0.09	0.02	0	54.2	-	-	-	-
fastText	36.52	36.09	10	28.7	35.65	34.95	10	38.3	0.41	0.13	0	48.6

Table 6: The method's performance on different word embedding representations and on new pairs of languages.

method did not converge to any reasonable solution for German and Finnish fastText embeddings, never exceeding 1% of accuracy. However, changing word2vec embeddings to fastText caused a significant increase in accuracy for Romance languages like Italian or Spanish, of even about 20% for the latter. It shows that fastText representation can be useful in cross-lingual mapping. Moreover, it suggests that the investigated method is not fully robust to changes of the word embedding algorithm.

The results for new language pairs demonstrate that Slavic languages are quite challenging for the cross-lingual mapping approach. The accuracy of Polish to English word2vec embedding mapping is the lowest of the reported ones, giving way only to the Polish-Spanish pair on which the method was able to converge only to a solution with an accuracy below 0.1%. The accuracy of fastText embedding mapping for new language pairs is also not particularly high. For the Czech-English pair, the method was never successful in learning a proper mapping, whereas reported accuracies for Polish-English and Polish-Spanish are significantly lower than those for English-Spanish or English-Italian pairs. Those results slightly undermine the claim of the method's robustness for not closely-related languages.

4.3. Unsupervised initialization with local search

The critical part of the mapping method is the unsupervised initialization, which is motivated by the assumption of isometry. As explained in Section 2.2., the assumption is that the similarity matrices for both mapped languages are equal except for the permutation of their columns and rows, so if we knew the correct translation pairs, we could order the rows and columns of a similarity matrix and obtain two equal matrices. It is clear that such a strong assumption does not hold exactly since that would mean that e.g., the languages have the same number of words or that they have the same number of synonyms for each concept. Nevertheless, this assumption, as the primary motivation to the heuristic approach described in Section 2.2., makes the method work. We decided to verify if this assumption can be used in a direct way to initialize the method. In our proposal, the initial dictionary is calculated by optimizing the following cost function

$$\min_{M'_Z \in Perm(M_Z)} ||M_X - M'_Z||_F$$

where $|| \cdot ||_F$ is the Frobenius norm and $Perm(M_Z)$ is the set of all possible matrices obtained by changing the order of rows and columns of M_Z. We optimize this function by a simple greedy search algorithm that starts from a random permutation and tries to obtain a better solution by exchanging places of two numbers in the permutation. The algorithm is run for an hour or until it converges to a local minimum. Then, its solution is the starting point for the mapping method described earlier.

The results of the experiments performed with this initialization method on dictionaries of 500 and 1000 word pairs are presented in Table 7. Besides the number of successful runs, best and averaged accuracy over ten runs, the table also shows the value of the cost function defined above. Note that this function is optimized only be the greedy approach, but the value of the function is also calculated on the permutation (dictionary) chosen by the original initialization method. According to the original paper, its initialization approach can be interpreted as a heuristic solution to the optimization problem defined earlier.

Overall, our initialization method performs worse than the original one, being competitive only on English-Finnish and English-Italian language pairs. The results of the cost function are, in our opinion, more interesting. One can see that even if the greedy approach finds a better solution in terms of the cost function, finally it does not obtain a better final result. Such a situation occurs for instance for English-Spanish pair on the dictionary of 1000 pairs. The value of cost function obtained by greedy is considerably lower than for the original method, meaning that the permutation found results in more similar matrices and better fits the isometry assumption. Nevertheless, the method initialized by this apparently-better permutation results in a mapping with almost 0% accuracy, whereas the heuristic initialization leads to a sensible translations.

A different situation is observed for English-Finish with 500 pairs used for initialization. Our initialization offers a solution with a higher cost function value than the original method, but greedy's results are more stable and even slightly more accurate. These results suggest that the assumption of isometry should be used rather carefully, and the development of new unsupervised initialization methods should not highly rely upon it. They also raise a question about the real reasons of the good performance of heuristic initialization proposed by Artetxe et al. (2018) as the original explanation by isometric assumption is questionable. Further research on that matter could also possibly explain the method's limited accuracy on some language pairs.

	500								1000							
	Original				Greedy				Original				Greedy			
	best	avg	s	cost	best	avg	s	cost	best	avg	s	cost	best	avg	s	cost
EN-ES	37.60	37.19	10	28.03	0.20	0.06	0	26.38	39.07	26.50	7	46.57	0.07	0.02	0	40.22
EN-FI	33.29	16.58	5	24.66	33.43	32.35	10	27.24	34.06	32.98	10	37.98	0.07	0.01	0	40.76
EN-DE	48.33	48.11	10	23.75	0.13	0.04	0	26.54	48.40	48.19	10	37.15	48.47	9.70	2	40.57
EN-IT	48.40	48.01	10	28.83	48.20	48.20	9	25.18	48.27	47.93	10	42.67	48.60	33.65	7	39.91
EN-PL	29.26	3.17	1	27.38	0.10	0.02	0	26.34	0.51	0.19	0	39.21	0.02	0.00	0	40.58

Table 7: Comparision of method's performance with different initialization strategies. "cost" denotes the the cost function value which is optimized by greedy algorithm.

5. Conclusion

In this work, we successfully perform a reproduction of experiments on the cross-lingual unsupervised word embedding mapping method presented by Artetxe et al. (2018). Although we confirm the results presented in this work, we demonstrate that the method presented there is not sufficiently robust to handle mappings between a bit more distant languages. Particularly, we show that some Slavic languages (Czech and Polish) are quite challenging for the method. Moreover, we expose the method's instability on some languages when changing the word embedding algorithm from word2vec to fastText. Søgaard et al. (2018b) shows that the isometry of embedding spaces is not preserved when embeddings are trained using different algorithms or datasets from different domains. Our work additionally raises doubts about the plausibility of isometric assumption when embeddings were trained using the same algorithm on a similarly collected dataset for both languages (i.e. from the same domain). Furthermore, we investigate the isometric assumption showing that the initialization that better complies with it does not always result in a better mapping.

6. Acknowledgements

This research was funded by Poznan Supercomputing and Networking Center. Mateusz Lango acknowledges also the support from the Institute of Computer Science Statutory Funds.

7. Bibliographical References

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 451–462, Vancouver, Canada, July. Association for Computational Linguistics.

Artetxe, M., Labaka, G., and Agirre, E. (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 1: Long Papers), pages 789–798, Melbourne, Australia, July. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017a). Word translation without parallel data. *CoRR*, abs/1710.04087.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017b). Word translation without parallel data. *CoRR*, abs/1710.04087.

Dinu, G., Lazaridou, A., and Baroni, M. (2014). Improving zero-shot learning by mitigating the hubness problem. *CoRR*, page arXiv:1412.6568, Dec.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

Hoshen, Y. and Wolf, L. (2018). Non-adversarial unsupervised word translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 469–478, Brussels, Belgium, October-November. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Ni, J., Dinu, G., and Florian, R. (2017). Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, Mar.

Smith, S. L., Turban, D. H. P., Hamblin, S., and Hammerla, N. Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *CoRR*, abs/1702.03859.

Søgaard, A., Ruder, S., and Vulic, I. (2018a). On the limitations of unsupervised bilingual dictionary induction. *CoRR*, abs/1805.03620.

Søgaard, A., Ruder, S., and Vulić, I. (2018b). On the limitations of unsupervised bilingual dictionary induction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 778–788, Melbourne, Australia, July. Association for Computational Linguistics.

Søgaard, A., Vulić, I., Ruder, S., Faruqui, M., and Hirst, G. (2019). Cross-Lingual Word Embeddings. Morgan Claypool.

Zhang, M., Liu, Y., Luan, H., and Sun, M. (2017). Adversarial training for unsupervised bilingual lexicon induction. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

8. Language Resource References

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Artetxe, M., Labaka, G., and Agirre, E. (2018). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In AAAI.

Dinu, G., Lazaridou, A., and Baroni, M. (2014). Improving zero-shot learning by mitigating the hubness problem. *CoRR*, page arXiv:1412.6568, Dec.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), pages 2214–2218, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Yamada, I., Asai, A., Shindo, H., Takeda, H., and Takefuji, Y. (2018). Wikipedia2vec: An optimized tool for learning embeddings of words and entities from wikipedia. *arXiv preprint 1812.06280*.

Zhang, M., Liu, Y., Luan, H., and Sun, M. (2017). Adversarial training for unsupervised bilingual lexicon induction. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.