Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

Article

Towards a Universal Semantic Dictionary

Maria Jose Castro-Bleda[1,†], Eszter Iklódi[2,‡], Gábor Recski[2,‡] and Gábor Borbély[2,‡]

[1] VRAIN Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Valencia, Spain; mcastro@dsic.upv.es

[2] Budapest University of Technology and Economics, Budapest, Hungary; eszter.iklodi@gmail.com, recski.gabor@aut.bme.h, borbely@math.bme.hu

* Correspondence: mcastro@dsic.upv.es

Abstract: A novel method for finding linear mappings among word embeddings for several languages, taking as pivot a shared, universal embedding space, is proposed in this paper. Previous approaches learn translation matrices between two specific languages, but this method learn translation matrices between a given language and a shared, universal space. The system was first trained on bilingual, and later on multilingual corpora as well. In the first case two different training data were applied; Dinu's English-Italian benchmark data, and English-Italian translation pairs extracted from the PanLex database. In the second case only the PanLex database was used. The system performs on English-Italian languages with the best setting significantly better than the baseline system of Mikolov et al. [1], and it provides a comparable performance with the more sophisticated systems of Faruqui and Dyer [2] and Dinu et al. [3]. Exploiting the richness of the PanLex database, the proposed method makes it possible to learn linear mappings among an arbitrary number of languages.

Keywords: Natural Language Processing; Semantics; Word embeddings; Multilingual embeddings; Translation; Artificial Neural Networks

---

1. Introduction

Computer-driven natural language processing plays an increasingly important role in our everyday life. In the current digital world, using natural language for human-machine communication has become a basic requirement. In order to meet this requirement, it is inevitable to analyze human languages semantically. Nowadays, state-of-the-art systems represent word meaning with high dimensional vectors, known as word embeddings.

Current embedding models are learned from monolingual corpora, and therefore infer language dependency. But one might ask if the structure of the different embeddings, i.e. different meaning representations, are universal among all human languages. Youn et al. [4] proposed a procedure for building graphs from concepts of different languages. They found that these graphs reflected a certain structure of meaning with respect to the languages they were built of. They concluded that the structural properties of these graphs are consistent across different language groups, and largely independent of geography, environment, and the presence or absence of literary traditions. Such findings led to a new research direction within the field of computational semantics, which focuses on the construction of universal meaning representations, most of the times in the form of cross-lingual word embedding models [5]. One way to create such models is to find mappings between embeddings of different languages [1,6,7].

Our work proposes a novel procedure for learning such mappings in the form of translation matrices that serve to map each language to a universal space. The method was first tested on bilingual, and later on multilingual corpora as well. With the bilingual experiments, we obtained on Dinu's benchmark data [3] a 0.377 precision@1 score for English-Italian and a 0.310 precision@1 score for Italian-English translation. These results, though, are far from the current state-of-the-art result on this dataset [7], but they are in the same order of magnitude or even better than many previous attempts [1–3]. For further bilingual and for some multilingual experiments an own dataset was created from the PanLex database [8]. We published the obtained scores of various experimental settings using this dataset [9]. Generally, bilingual experiments using only the PanLex dataset resulted in worse scores than using only Dinu's dataset, but combining the two showed a slight improvement in the Italian-English direction. Multilingual experiments were carried out using three different languages, English, Italian, and Spanish, at the same time. The obtained pairwise precision values showed worse results, than when the system was trained in bilingual mode. However, these results are still promising considering that a completely new approach was implemented, and they showed that the system definitely learned from a data which is available for a wide range of languages.

Section 2 summarizes the progress made on learning translation matrices between word embeddings over the last five years. Section 3 discusses the proposed method in detail. Following that, Section 4 describes our experimental setup, and Sections 5 and 6 report and analyze the obtained results. Finally, Section 7 concludes the advantages and disadvantages of the proposed model, and also discusses some improvements for future work.

2. Related work

2.1. Word embeddings

One way to build semantic representations is to use distributional models. The idea is based on the observation that synonyms or words with similar meanings tend to occur in similar contexts, or as it was phrased by Firth in 1957: "You shall know a word by the company it keeps" [10]. For example, in the following two sentences *"The cat is walking in the bedroom"* and *"A dog was running in a room"* words like *"dog"* and *"cat"* have exactly the same semantic and grammatical roles, therefore we could easily imagine the two sentences in the following variations: *"The dog is walking in the bedroom"* and *"A cat was running in a room"* [11]. Based on this intuition, what distributional models are aiming to do is to compute the meaning of a word from the distribution of words around it [12]. The obtained meaning representations are usually high dimensional vectors, called word embeddings, which refer to their characteristic feature that they model a world by embedding it into a vector space.

2.2. Monolingual word embeddings

Mikolov et. al [13] suggested a Bag-of-words Neural Network, more specifically two architectures, for learning monolingual word embeddings. The first one, denoted as the Continuous Bag-of-Words Model (CBOW) tried to predict the current word based on the context, whereas the second one, denoted as the continuous skip-gram model tried to maximize the classification of a word based on another word in the same sentence. The CBOW turned out to be slightly better on syntactic tasks and the skip-gram on semantic tasks. Mikolov's procedure has become known as the *word2vec*[1] procedure.

2.3. Multilingual word embeddings

In 2013, Mikolov et al. [1] published a simple two-step procedure for creating universal embeddings. In the first step they built monolingual models of languages using huge corpora, and in

[1] http://deeplearning4j.org/word2vec

the second step a small bilingual dictionary was used to learn linear projection between the languages. The optimization problem was the following:

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

where W denotes the transformation matrix, and $\{x_i, z_i\}_{i=1}^{n}$ are the continuous vector representations of word translation pairs, with x_i being in the source language space and z_i in the target language space.

Faruqui and Dyer [2] proposed a procedure to obtain multilingual word embeddings by concatenating the two word vectors coming from the two languages, applying Canonical Correlation Analysis. Xing et al. [14] found that bilingual translation can be largely improved by normalizing the embeddings and by restricting the transformation matrices into orthogonal ones. Dinu et al. [3] showed that the neighborhoods of the mapped vectors are strongly polluted by hubs, which are vectors that tend to be near a high proportion of items. They proposed a method that computes hubness scores for target space vectors and penalizes those vectors that are close to many words, i.e. hubs are down-ranked in the neighboring lists. Lazaridou et al. [15] studied some theoretical and empirical properties of a general cross-space mapping function, and tested them on cross-linguistic (word translation) and cross-modal (image labelling) tasks. They also introduced the use of negative samples during the learning process. Amar et al. [16] proposed methods for estimating and evaluating embeddings of words in more than fifty languages in a single shared embedding space. Since English usually offers the largest corpora and bilingual dictionaries, they used the English embeddings to serve as the shared embedding space. Artetxe et al. [17] built a generic framework that generalizes previous works made on cross-linguistic embeddings and they concluded that the best systems were the ones with orthogonality constraint and a global pre-processing with length normalization and dimension-wise mean centering. Smith et al. [6] also proved that translation matrices should be orthogonal, for which they applied Singular Value Decomposition (SVD) on the transformation matrices. Besides, they also introduced a novel "inverted softmax" method for identifying translation pairs. All these works listed above applied supervised learning. However, in 2017 Conneau et al. [7] introduced an unsupervised way for aligning monolingual word embedding spaces between two languages without using any parallel corpora. This unsupervised procedure holds the current state-of-the-art results on Dinu's benchmark word translation task.

3. Proposed method

In summary, this work proposes a novel method for learning linear mappings between word translation pairs in the form of translation matrices. These translation matrices learn to map pre-trained word embeddings into a universal vector space. During training the cosine similarity of word translation pairs is maximized, which is calculated in the universal space. After mapping the embeddings of two different languages into this universal space, the cosine similarity of the actual translation pairs should be high. At test time the system is evaluated with the precision metric, principally used for word translation tasks.

3.1. Cosine similarity and precision

Cosine similarity[2] is a measure of similarity between two non-zero vectors. It is calculated as the normalized dot product of two vectors, as shown in Equation 2. In fact, cosine similarity is a space that measures the cosine of the angle of two vectors. It is important to note that cosine similarity is not a proper distance metric, since the triangle inequality property does not apply. In word similarity

[2] https://en.wikipedia.org/wiki/Cosine_similarity

tasks, however, this metric is used for measuring the similarity of two words represented as word vectors. Although cosine similarity values by definition are in range of [-1, 1], in word similarity tasks it is particularly used in positive space, [0, 1], where parallel vectors are similar and orthogonal vectors are dissimilar.

$$cos\_sim = \cos\theta = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} \tag{2}$$

Precision is a metric used for measuring the performance of translator systems, which intend to learn to translate from a source language into a target language. On the target side a look-up space is defined, which could, for example, correspond to the most frequent 200K words of the target language, as in our experiments. After translating a word, the N word vectors of the look-up space that are closest to the translated one are regarded. The Precision @N metric denotes the percentage of how many times the real translation of a word is found among the N closest word vectors in the look-up space. Usual N values are 1, 5, and 10.

3.2. The objective function

The objective of the proposed method is to learn linear mappings in the form of translation matrices that are obtained by maximizing the cosine similarity of gold word translation pairs in a universal space. Therefore, for each language one single translation matrix is searched that maps the language from its original vector space to the universal one. The method tries to bring the translation pairs close together in a shared, universal space. Therefore, it is not only applicable for language pairs but for any number of languages as well. The main advantage is that by introducing new languages the number of the learned parameters remains linear to the number of languages since instead of learning pair-wise translation matrices, for each language only one matrix is learned, the one that maps directly to this shared, universal space.

Let L be a set of languages, and TP a set of translation pairs where each entry is a tuple of two in the form of (w_1, w_2) where w_1 is a word in language L_1 and w_2 is a word in language L_2, and both L_1 and L_2 are in L. Then, let's consider the following equation to optimize:

$$\frac{1}{|TP|} \cdot \sum_{\substack{L_1,L_2 \\ \in L}} \sum_{\substack{(w_1,w_2) \\ \in TP}} cos\_sim(w_1 \cdot T_1, w_2 \cdot T_2) \tag{3}$$

where T_1 and T_2 are translation matrices mapping L_1 and L_2 to the universal space. Since the equation is normalized with the number of translation pairs in the TP set, the optimal value of this function is 1. Off-the-shelf optimizers are programmed to find local minimum values, so during the training process the loss function is multiplied by -1. Word vectors are always normalized, so the cos_sim reduces to a simple dot product.

At test time, first, both source and target language words are mapped into the universal space, and from the most frequent 200k mapped target language words a look-up space is defined. Then, the system is evaluated with the Precision metric, more specifically with Precision @1, @5, and @10. The distance assigned to the word vectors when searching in the look-up space is the cos_sim.

Previous works, such as Mikolov et al. [6] or Conneau et al. [7], suggested restricting the transformation matrix to an orthogonal one. From an arbitrary transformation matrix T an orthogonal T' can be obtained by applying the SVD procedure. Our experiments showed that by applying SVD on the transformation matrices the learning is significantly faster. Best results were obtained when applying the SVD only once, at the beginning of the learning process.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 July 2019 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at *Appl. Sci.* **2019**, *9*, 4060; doi:10.3390/app9194060

4. Experimental setup

4.1. Pre-trained word embeddings

For pre-trained word embeddings we took the *fastText* embeddings published by Conneau et al. [7]. These embeddings were trained by applying their novel method where words are represented as a bag of character n-grams [18]. This model outperformed Mikolov's [13] `CBOW` and `skipgram` baseline systems that did not take any sub-word information into account. Conneau's pre-trained word vectors trained on Wikipedia are available for 294 languages[3].

Some experiments were also run by using the same embedding that was used by Dinu et al. [3] in their experiments. These word vectors were trained with *word2vec* and then the 200K most common words in both the English and Italian corpora were extracted. The English word vectors were trained on the WackyPedia/ukWaC and BNC corpora, while the Italian word vectors were trained on the WackyPedia/itWaC corpus. This word embedding will be referred to as the *WaCky* embedding.

4.2. English-Italian setup of Dinu

Dinu et.al [3] constructed an English-Italian gold dictionary split into a training and a test set that is now being used as benchmark data for evaluating English-Italian word translation tasks. Both training and test translation pairs were extracted from a dictionary built from Europarl Eng-Ita[4] [19].

For the test set they used 1500 English words split into 5 frequency bins, 300 randomly chosen in each bin. The bins were defined in terms of rank in the frequency-sorted lexicon: [1-5K], [5K-20K], [20K-50K], [50K-100K], and [100K-200K]. Some of these 1500 English words had multiple Italian translations in the Europarl dictionary, so the resulting test set contained 1869 word pairs all together, with 1500 different English, and with 1849 different Italian words (see Table 1).

For the training set, the above-mentioned Europarl dictionary was first sorted by the English frequency. Then the top 5K entries were extracted and care was taken to avoid any overlap with the test elements on the English side. On the Italian side, however, an overlap of 113 words was still present. In the end, the training set contained 5K word pairs with 3442 different English, and 4549 different Italian words (see Table 1).

Table 1. Statistics of word counts.

Set	Language	No. words
Train (5000 word pairs)	Eng.	3442
	Ita.	4549
Test (1869 word pairs)	Eng.	1500
	Ita.	1849

4.3. The PanLex Corpus

PanLex [8] is a nonprofit organization that aims to build a multilingual lexical database from available dictionaries in all languages. The name PanLex is coming from the words *panlingual* and *lexical*, which reflect the main objective of this project. They are basically digitizing and centering the content of different, already existing dictionaries made by domain experts. Own translations are not accepted. To each translation pair a confidence value is assigned, which can be used for filtering the extracted data. These confidence values are in the range of [1, 9], with 9 meaning high and 1 meaning low confidentiality. The main purpose is to preserve the diversity of languages, so the collection

3 http://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md
4 http://opus.lingfil.uu.se/

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 July 2019 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

Table 2. Sample of PanLex entries of the extracted tsv file.

English	Italian	Confidence values
Sarajevo	Sarajevo	9
euro	euro	9
simple	semplice	8
difficult	difficile	8
college	università	7
plausible	verisimile	7
sea	mare	6
sky	cielo	6
better	meglio	5
inform	informare	5
combustible	combustibile	4
office	ufficio	4
sorcerer	conscitore	3
it	ella	3
Great Wall of China	Grande muraglia cinese	2
factory workers	lavoratori dell'industria	2
stay	restare	1
sometimes	qualche volta	1

of "threatened" or "endangered" languages and dictionaries of rare language combinations are top priority. Some examples of the English-Italian PanLex data can be seen in Table 2.

PanLex also exhibits different *language varieties* that include, among others, regional variations and different writing systems. A *language variety* is denoted with a three-letter *language code*, e.g. `eng` for English, and with a three-digit *variety code*, e.g. `000`. To the most widely spoken variety of a language usually the `000` *variety code* is assigned. When extracting data from the PanLex database, in all cases, the *language variety* with the smallest *variety code* was taken.

4.4. Dataset creation from PanLex

The procedure applied for extracting a proper data from the PanLex database for training multilingual embedding models roughly follows the same steps as in [3]. After extracting the raw translation pairs form the PanLex database, a filtered version of entries was formed by dropping translations with a confidence value below 7 and those for which no word vector was found in the *fastText* embedding. This results in an English-Italian word translation set containing 69,623 entries.

For the test set 1500 English words were taken and split into 5 frequency bins, 300 randomly assigned to each bin. The bins were defined the same way as in [3], i.e., in terms of rank in the frequency-sorted lexicon: [1-5K], [5K-20K], [20K-50K], [50K-100K], and [100K-200K]. In [7], the word vectors sorted by their frequency in descending order were published, and this order was used as the source of English word frequency data. In the PanLex database it is a common issue that one English word has sometimes as many as 10 different Italian translations. Therefore, in order to avoid having an undesirably huge test set with many Italian synonyms only those English words were selected, for which in the corresponding bin only one Italian translation was present. This way the obtained test set contains exactly 1500 word pairs, which are made up of 1500 different English words and their Italian translations.

For the training set, the 69,623 entries were first sorted by their English frequency, then the top 5K entries were extracted and, as in Dinu *et al.*, care was taken to avoid any overlap with test elements on the English side. Then, the top 5K entries were selected in three different ways:

1. Simply the first 5K entries were taken.
2. The first 5K different English words were taken with the most frequent Italian translation.
3. Only those English words were taken for which only one Italian translation was present.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 July 2019 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

4.5. Baseline experimental setting

For the baseline system the *fastText* embedding was used as a pre-trained embedding and the system was trained on Dinu's English-Italian data. For parameter adjustment Dinu's training data was split into train and validation sets such that no overlap was present on the English side, i.e. no word appeared in both sets; this follows Dinu's procedure of constructing their original training and test sets. It should be noted that this does not apply for Italian words. For the word count and overlap statistics of Dinu's original training and test sets see Table 3 and for the same statistics of the newly produced training and validation sets see Table 4.

Table 3. Statistics of the original train and test split of Dinu's data.

Number of English words	train	3442
Number of Italian words	train	4549
Number of English words	test	1500
Number of Italian words	test	1849
Overlap English		0
Overlap Italian		113

Table 4. Statistics of the new train and validation split of Dinu's data.

Number of English words	train	3098
Number of Italian words	train	4129
Number of English words	valid	344
Number of Italian words	valid	499
Overlap English		0
Overlap Italian		80

The system was adjusted on the previously described training and validation split. For the optimizer the tensorflow implementation[5] of the Adagrad algorithm [20] was used. For evaluation the most frequent 200K words of the target space embedding were used as look-up space for calculating Precision @1, @5, and @10. In all cases both English-Italian and Italian-English precision scores were observed. In addition, the average cosine similarity value of the validation set was also checked. During training and validation as well the precision and similarity values were all calculated in the universal space. Gold dictionaries were constructed from the input data files themselves. Following Dinu, any word appearing in the dictionary was considered a valid translation. Various translations may come from synonyms or different male-female forms on the Italian side.

5. Experimental results

5.1. Parameter adjustment using Dinu's data

First, parameter adjustment was performed using Dinu's data, which gave 0.1 as the best learning rate and 64 as the best batch size, where batch size is equal to the number of translation pairs used in one iteration. With applying SVD only once at the beginning the obtained results of our best system are significantly worse than state-of-the-art results on this benchmark data, but they are comparable with or even better than some of the previous models discussed in Section 2.

[5] https://www.tensorflow.org/api_docs/python/tf/train/AdagradOptimizer

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 July 2019 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

5.2. Experimenting with SVD

Previous works, such as [6] or [7], suggested restricting the transformation matrix to an orthogonal one. Based on these findings this system also features a configuration option of applying an SVD. Three different SVD modes were studied:

- **0**: Not using SVD at all
- **1**: Using SVD after every n-th epoch
- **2**: Using SVD only once, at the beginning

In the following experiments the same datasets were used as for parameter adjustment. Learning rate was set to 0.1 and batch size to 64 as found the best setup before. Altogether 200 epochs were done and evaluation was performed on every 10-th epoch.

5.2.1. Not using SVD

This experiment was carried out without applying any SVD. Translation matrices were initialized with random numbers. Figure 1 shows that similarity values are monotone increasing, meaning that the system is learning. But the learning process is relatively slow since even after 200 epochs the similarity score is still quite low, bearing in mind that the optimal value is 1.0.

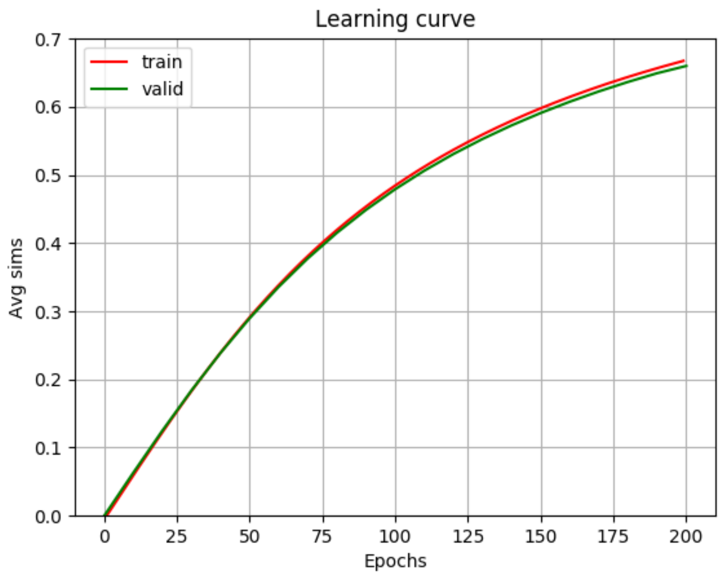

Figure 1. Learning curve of experimenting without using SVD (svd_mode = 0).

5.2.2. SVD after every n-th epoch

This experiment was carried out applying SVD several times over the whole learning process. SVD was made on every 50-th epoch, i.e. 4 times altogether. Figure 2 shows how the learning curve breaks down every time after applying an SVD on the translation matrices, and, also, how fast it is back once again to the previous high similarity values. Besides, this time the average cosine similarity score was higher even at the beginning than it was after 200 epochs with the previous setting, where no SVD was done. Applying SVD on the transformation matrices seems to accelerate the learning process significantly. The learning curve also shows that SVD-to-SVD fractions have exactly the same trajectory regardless of the number of previous epochs done.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 July 2019 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

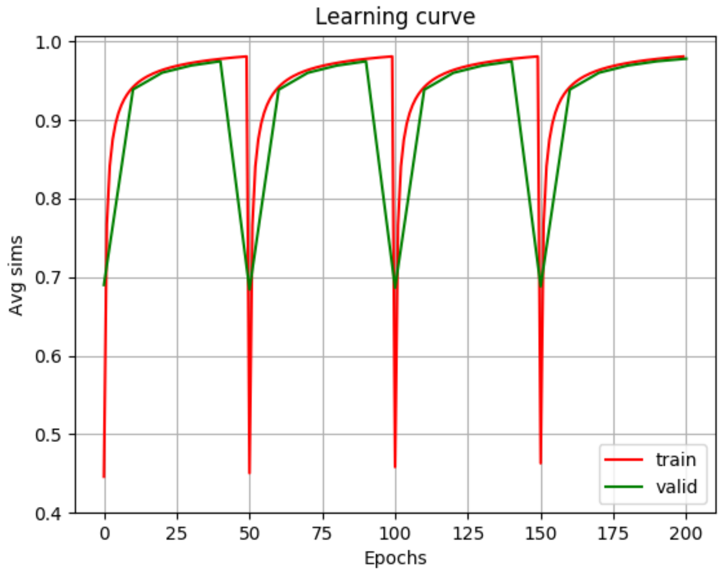

Figure 2. Learning curve of experimenting with SVD after every n-th epoch (svd_mode = 1).

5.2.3. SVD at the beginning

This experiment was carried out applying SVD only once, at the very beginning. This means, in simple terms, that instead of a random initial transformation matrix, the system tried to adjust an orthogonal one. Figure 3 shows that the learning curve is monotone increasing, and owing to the initial SVD it gets fairly high right at the beginning.

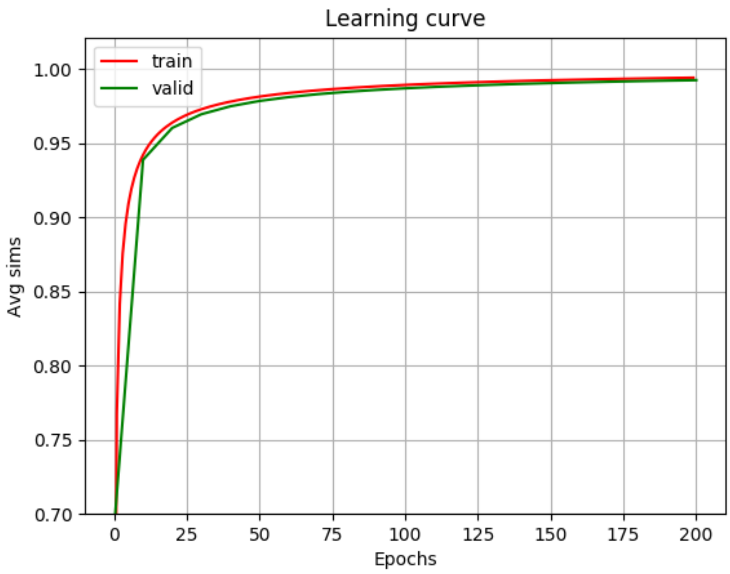

Figure 3. Learning curve of experimenting with SVD at the beginning (svd_mode = 2).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 July 2019 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

5.3. Experiments with the PanLex data

5.3.1. Comparing different dataset construction methods

Tables 5 and 6 compare the results of different dataset construction methods. It is important to note that in the first case the English word of the 5000-th translation pair is only the 845-th most frequent English word, meaning that there is only 845 different English words in the training set and that, on average, there is 5-6 different Italian translations to each of them. In the second case, where every English word is kept but only with the most frequent Italian translation, this number is 9007. In the last case, however, the 5000-th entry is made up of the 39426-th most frequent English and the 31543-th most frequent Italian words. Still, this last training set provides the best results, so for further experiment this construction method was applied.

Table 5. English-Italian precision values with the different training sets.

Precision	@1	@5	@10
First 5K entry	0.0093	0.0253	0.0367
First 5K English words with retaining one translation	0.1120	0.2073	0.2427
First 5K English words with one translation	**0.1960**	**0.3087**	**0.3440**

Table 6. Italian-English precision values with the different training sets.

Precision	@1	@5	@10
First 5K entry	0.0000	0.0007	0.0007
First 5K English words with retaining one translation	0.1114	0.2052	0.2440
First 5K English words with one translation	**0.1838**	**0.3059**	**0.3443**

5.3.2. Experimenting with different training set sizes

Table 7 summarizes the results of experiments with different training set sizes. The 3K dataset proved to be the best on the English-Italian translation, but on the Italian-English it is only slightly better, than the 5K dataset. This behaviour of performing better on the smaller training sets is fairly understandable since as a consequence of the way the training set was constructed, as we are taking in more and more entries, we are actually taking in less and less frequent English words and their Italian translations, for which words neither the embedding nor the translations are precise enough. Since Dinu's benchmark data contains 5K entries in the training set, despite the slightly worse performance we kept using the 5K dataset for the sake of comparability with other result.

Table 7. Experiments with different training set sizes

	Eng-Ita			**Ita-Eng**		
Prec.	@1	@5	@10	@1	@5	@10
1K	0.1500	0.2847	0.3340	0.1391	0.2761	0.3256
3K	**0.2127**	**0.3473**	**0.3933**	**0.2232**	**0.3650**	**0.4152**
5K	0.1980	0.3193	0.3620	0.2212	0.3555	0.4030
10K	0.1613	0.2807	0.3227	0.1879	0.3012	0.3372

5.4. Comparison of systems trained on Dinu's and PanLex data

In the next step, some experiments were made to determine which data is more apt for learning linear mappings between embeddings. In order to compare all the experiments objectively subsets of the original test sets were created. These subsets do not contain any English word present either in the Dinu training set or in the PanLex training set. Table 8 summarizes the number of word pairs in the old and the new test sets. It should be noted that by this reduction mainly the most

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 July 2019 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

common English words are affected, and therefore worse scores are expected compared to the previous train-on-Dinu-test-on-Dinu, or train-on-PanLex-test-on-PanLex top results. Scores on Dinu's test set are shown in Table 9 and on the PanLex data in Table 10. The obtained results show that training on the PanLex data cannot beat the system trained on Dinu's data, which performs better both on Dinu's and on the PanLex test sets. Not even combining the two training sets succeeds in achieving significantly better results, although on the PanLex test set it does improve the scores in the Italian-English direction.

Table 8. Word reduction of the new test sets

Test set	No. word pairs in old	No. word pairs in new
Dinu	1869	1455
PanLex	1500	1242

Table 9. Comparing Dinu's and PanLex data on Dinu's test set

	Eng-Ita			Ita-Eng		
Precision	@1	@5	@10	@1	@5	@10
Train:Dinu - Test:old	0.3770	0.5647	0.6245	0.3103	0.5018	0.5474
Train:Dinu - Test:new	**0.3560**	**0.5407**	**0.5978**	**0.2917**	**0.4792**	**0.5215**
Train:PanLex - Test:new	0.1360	0.2309	0.2594	0.1361	0.2556	0.2965
Train:Dinu+PanLex - Test:new	0.2930	0.4349	0.4861	0.2910	0.4556	0.5090

Table 10. Comparing Dinu's and PanLex data on the PanLex test set

	Eng-Ita			Ita-Eng		
Precision	@1	@5	@10	@1	@5	@10
Train:PanLex - Test:old	0.1960	0.3087	0.3440	0.1838	0.3059	0.3443
Train:PanLex - Test:new	0.1812	0.2858	0.3196	0.1668	0.2835	0.3213
Train:Dinu - Test:new	**0.2295**	**0.4171**	**0.4839**	0.2227	0.3763	0.4199
Train:Dinu+PanLex - Test:new	0.2295	0.3712	0.4275	**0.2498**	**0.4026**	**0.4495**

5.5. Continuing the training with PanLex data

Another experiment was conducted to continue the baseline system trained on Dinu's data with the PanLex data. In other words, it is the same as initializing the translation matrices of the PanLex training process with previously learned ones. The baseline system reaches its best performance between 2000 and 4000 epochs, depending on which precision value is regarded. Table 11 shows that on the English-Italian task there is no improvement at all, while on the Italian-English task with the best setting slightly better scores are achieved on precision @1 and @10 values.

Table 11. Continuing the baseline system with the PanLex data.

	Eng-Ita			Ita-Eng		
Precision	@1	@5	@10	@1	@5	@10
Original	**0.3770**	**0.5647**	**0.6245**	0.3103	**0.5018**	0.5474
Cont. from 2000	0.3426	0.5256	0.5802	**0.3229**	0.4882	**0.5535**
Cont. from 3000	0.3535	0.5416	0.5970	0.3229	0.4840	0.5465
Cont. from 4000	0.3510	0.5273	0.5911	0.3118	0.4701	0.5243

5.6. Experiments using three languages

Finally, a multilingual experiment was carried out where the system was trained on three languages - English, Italian, and Spanish - at the same time. During training the system learns three different translation matrices, one for English-universal, one for Italian-universal, and one for Spanish-universal space mapping. For example, in order to learn the English-universal translation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 July 2019 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

matrix, both the English-Italian and the English-Spanish dictionaries are used, according to Equation (3). Batches are homogeneous, but two following batches are always different in terms of the language origins of the contained data. That is, first an English-Italian batch is fed to the system, then an English-Spanish batch, after that an Italian-Spanish batch, and so on. First, bilingual models were trained in order to compare them later with the multilingual system. The results of the bilingual models are summarized in Table 12. Results are best on the Italian-Spanish task. Next, the system was trained using all the three languages at the same time. During the training process the model was evaluated on the bilingual test datasets of which the results are shown in Table 13. The obtained results show that no advantage was achieved by extending the number of languages, since the multilingual model performs worse than any of the pairwise bilingual models.

Table 12. Results of bilingual models trained pairwise on the three different languages.

	L1-L2			**L2-L1**		
Precision	@1	@5	@10	@1	@5	@10
Eng-Ita	0.2080	0.3280	0.3687	0.2082	0.3386	0.3904
Eng-Spa	0.2840	0.4320	0.4800	0.2883	0.4331	0.4836
Spa-Ita	0.3920	0.5340	0.5813	0.3655	0.5291	0.5750

Table 13. Bilingual results of the multilingual model trained using three different languages at the same time.

	L1-L2			**L2-L1**		
Precision	@1	@5	@10	@1	@5	@10
Eng-Ita	0.1573	0.2667	0.3127	0.1638	0.2942	0.3386
Eng-Spa	0.1947	0.2973	0.3447	0.2350	0.3538	0.4064
Spa-Ita	0.2520	0.3640	0.4160	0.2568	0.3723	0.4162

6. Comparison of the experiments

Tables 14 and 15 show our results on Dinu's dataset compared to other published works. Our results are worse than those current state-of-the-art, but they are still comparable or even better than several of previous attempts. The advantage of the proposed method compared to other procedures is that it is applicable for an arbitrary number of languages at the same time. Though the multilingual experiments on the PanLex dataset showed worse results than the bilingual ones, they are still showing convergence and can serve as a baseline for future multilingual experiments.

Table 14. Comparing English-Italian results on Dinu's data.

Eng-Ita	**@1**	**@5**	**@10**
Mikolov et al. (2013) [1]	0.338	0.483	0.539
Faruqui et al. (2014) [2]	0.361	0.527	0.581
Dinu et al. (2014) [3]	0.385	0.564	0.639
Smith et al. (2017) [6]	0.431	0.607	0.651
Conneau et al. (2017) [7]	0.662	0.804	0.834
Proposed method	0.377	0.565	0.625

 doi:10.20944/preprints201907.0336.v1

Peer-reviewed version available at Appl. Sci. 2019, 9, 4060; doi:10.3390/app9194060

Table 15. Comparing Italian-English results on Dinu's data.

Ita-Eng	@1	@5	@10
Mikolov et al. (2013) [1]	0.249	0.410	0.474
Faruqui et al. (2014) [2]	0.310	0.499	0.570
Dinu et al. (2014) [3]	0.246	0.454	0.541
Smith et al. (2017) [6]	0.380	0.585	0.636
Conneau et al. (2017) [7]	0.587	0.765	0.809
Proposed method	0.310	0.502	0.547

7. Conclusions and future work

This paper proposes a novel method for finding linear mappings between word embeddings in different languages. As a proof of concept a framework was developed which enabled basic parameter adjustments and flexible configuration for initial experimentation.

An interesting finding was that the system learned much faster when an initial SVD was applied on the translation matrices. Results obtained with these settings on Dinu's data showed that the proposed model did learn from the data. The obtained precision scores, though, were far from current state-of-the-art results on this benchmark data, they were comparable with results of previous attempts. The proposed model performed much better using the *fastText* embeddings [7], than using Dinu's WaCky embeddings [3].

Thereafter, an English-Italian dataset was extracted from the PanLex database, from which training and test datasets were constructed roughly following the same steps that Dinu et al. [3] took. The system was trained and tested on both Dinu's and PanLex test sets, and in both cases the matrices trained on Dinu's data were the ones reaching higher scores. On the PanLex data experiments with different training set sizes were executed, out of which the 3K training set gave the best results. Continuing the training of the matrices obtained by using Dinu's data with the PanLex dataset brought a slight improvement on the Italian-English scores, but English-Italian scores only got worse.

Finally, the system was trained on three different languages at the same time. The obtained pairwise precision values are proved to be worse than the results obtained when the system was trained in bilingual mode. However, these results are still promising considering that a completely new approach was implemented, and they showed that the system definitely learned from a data which is available for a wide range of languages.

The approach is quite promising but in order to reach state-of-the-art performance the system has to deal with some mathematical issues, for example dimension reduction in the universal space. Further experimentation in multilingual mode with an extended number of languages could also provide meaningful outputs. By involving expert linguistic knowledge various sets of languages could be constructed using either only very close languages, or, on the contrary, using very distant languages. Thanks to the PanLex database, bilingual dictionaries can easily be extracted, which can, then, be directly used for multilingual experiments.

Funding: This research was funded by Spanish MINECO and FEDER grant number TIN2017-85854-C4-2-R.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

SVD Singular Value Decomposition

Peer-reviewed version available at *Appl. Sci.* **2019**, *9*, 4060; doi:10.3390/app9194060

References

1. Mikolov, T.; Le, Q.V.; Sutskever, I. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* **2013**.
2. Faruqui, M.; Dyer, C. Improving vector space word representations using multilingual correlation. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, pp. 462–471.
3. Dinu, G.; Lazaridou, A.; Baroni, M. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568* **2014**.
4. Youn, H.; Sutton, L.; Smith, E.; Moore, C.; Wilkins, J.F.; Maddieson, I.; Croft, W.; Bhattacharya, T. On the universal structure of human lexical semantics. *Proceedings of the National Academy of Sciences* **2016**, *113*, 1766–1771.
5. Ruder, S.; Vulić, I.; Søgaard, A. A survey of cross-lingual word embedding models. *arXiv preprint arXiv:1706.04902* **2017**.
6. Smith, S.L.; Turban, D.H.; Hamblin, S.; Hammerla, N.Y. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859 (publised at ICRL2017)* **2017**.
7. Conneau, A.; Lample, G.; Ranzato, M.; Denoyer, L.; Jégou, H. Word translation without parallel data. *arXiv preprint arXiv:1710.04087* **2017**.
8. Kamholz, D.; Pool, J.; Colowick, S.M. PanLex: Building a Resource for Panlingual Lexical Translation. LREC, 2014, pp. 3145–3150.
9. Eszter, I.; Recski, G.; Borbély, G.; Castro-Bleda, M.J. Building a global dictionary for semantic technologies. Proc. Iberspeech, 2018, pp. 286–290. doi:10.21437/IberSPEECH.2018-60.
10. Firth, J.R. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis* **1957**.
11. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *Journal of machine learning research* **2003**, *3*, 1137–1155.
12. Jurafsky, D.; Martin, J.H. *Speech and language processing*; Pearson London:, 2017.
13. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* **2013**.
14. Xing, C.; Wang, D.; Liu, C.; Lin, Y. Normalized word embedding and orthogonal transform for bilingual word translation. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 1006–1011.
15. Lazaridou, A.; Dinu, G.; Baroni, M. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, Vol. 1, pp. 270–280.
16. Ammar, W.; Mulcaire, G.; Tsvetkov, Y.; Lample, G.; Dyer, C.; Smith, N.A. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925* **2016**.
17. Artetxe, M.; Labaka, G.; Agirre, E. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 2289–2294.
18. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* **2016**.
19. Tiedemann, J. Parallel Data, Tools and Interfaces in OPUS. LREC, 2012, pp. 2214–2218.
20. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* **2011**, *12*, 2121–2159.