IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

Building a global dictionary for semantic technologies

Eszter Iklódi[1], *Gábor Recski*[1], *Gábor Borbély*[2], *Maria Jose Castro-Bleda*[3]

[1]Department of Automation and Applied Informatics, Budapest University of Technology and Economics, Budapest, Hungary
[2]Department of Algebra, Budapest University of Technology and Economics, Budapest, Hungary
[3]Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, Valencia, Spain

eszter.iklodi@gmail.com, recski.gabor@aut.bme.hu, borbely@math.bme.hu, mcastro@dsic.upv.es

Abstract

This paper proposes a novel method for finding linear mappings among word vectors for various languages. Compared to previous approaches, this method does not learn translation matrices between two specific languages, but between a given language and a shared, universal space. The system was trained in two different modes, first between two languages, and after that applying three languages at the same time. In the first case two different training data were applied; Dinu's English-Italian benchmark data [1], and English-Italian translation pairs extracted from the PanLex database [2]. 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. [3], and it provides a comparable performance with the more sophisticated systems of Faruqui and Dyer [4] and Dinu et al. [1]. Exploiting the richness of the PanLex database, the proposed method makes it possible to learn linear mappings among an arbitrary number languages.

Index Terms: 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. [5] 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 [6]. One way to create such models is to find mappings between embeddings of different languages [3, 7, 8]. 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.

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 the experimental setup we used and reports the obtained results. Finally, Section 5 concludes the advantages and disadvantages of the proposed model, and also discusses some improvements for future work.

2. Related work

In 2013, Mikolov et al. [3] 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 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 [4] proposed a procedure to obtain multilingual word embeddings by concatenating the two word vectors coming from the two languages, applying canonical correlation analysis (CCA). Xing et al. [9] found that bilingual translation can be largely improved by normalizing the embeddings and by restricting the transformation matrices into orthogonal ones. Dinu et al. [1] showed that the neighbourhoods 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 neighbouring lists. Lazaridou et al. [10] 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. [11] proposed methods for estimating and evaluating embeddings of words in more than fifty languages in a single shared embedding space. Since

10.21437/IberSPEECH.2018-60

English usually offers the largest corpora and biligual dictionaries, they used the English embeddings to serve as the shared embedding space. Artetxe et al. [12] 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. [7] 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. [8] 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. For comparing the different results see Table 1 and Table 2.

3. Proposed method

We propose a method that learns linear mappings between word translation pairs in the form of translation matrices that 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. The method is applicable for any number of languages. Since, independent of the number of languages applied during training, for each language always exactly one translation matrix is learned, by introducing new languages, the number of the learned parameters remains linear to the number of the applied languages.

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:

$$\frac{1}{|TP|} \cdot \sum_{\substack{L_1,L_2\\ \in L}} \sum_{\substack{(w_1,w_2)\\ \in TP}} cos\_sim(w_1 \cdot T_1, w_2 \cdot T_2) \qquad (2)$$

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, where Precision @N denotes the percentage of how many times the real translation of a source word is found among the N closest word vectors in the look-up space. 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. [7] or Conneau et al. [8], 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.

4. Experiments

4.1. Experimental setup

4.1.1. Pre-trained word embeddings

For pre-trained word embeddings we took the *fastText* embeddings published by Conneau et al. [8]. These embeddings were trained by applying their novel method where words are represented as a bag of character n-grams [13]. This model outperformed Mikolov's [14] `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[1].

Some experiments were also run by using the same embedding that was used by Dinu et al. [1] 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.

4.1.2. Gold dictionaries

First, we ran the experiments on Dinu's English-Italian benchmark data [1]. It is an English-Italian gold dictionary split into a train and a test set, which was built from Europarl en-it[2] [15]. For the test set they used 1,500 English words split into 5 frequency bins, 300 randomly chosen in each bin. The bins are 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 have multiple Italian translations in the Europarl dictionary, so the resulting test set contains 1869 word pairs all together, with 1500 different English, and with 1849 different Italian words. For the training set the top 5k entries were extracted and care was taken to avoid any overlap with test elements on the English side. On the Italian side, however, an overlap of 113 words is still present. In the end the train set contains 5k word pairs with 3442 different English, and 4549 different Italian words.

Then, we built another golden dictionary similar to that of Dinu's, but this time the translation pairs were extracted from the PanLex [2] database. PanLex is a nonprofit organization that aims to build a multilingual lexical database from available dictionaries made by domain experts in all languages. 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 confidence. During the extraction process, translations with a confidence value below 7 and those for which no word vector was found in the *fastText* embedding were dropped. Then, training and test sets were constructed following Dinu's steps, except for that only those English words were taken for which only one Italian translation was present. Experiments showed that otherwise a serious noise was brought into the system, since in many cases one English word might have up to 10 different Italian translations.

[1] http://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[2] http://opus.lingfil.uu.se/

4.2. Results

4.2.1. 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. For comparison see Table 1 and Table 2.

Eng-Ita	@1	@5	@10
Mikolov et al.	0.338	0.483	0.539
Faruqui et al.	0.361	0.527	0.581
Dinu et al.	0.385	0.564	0.639
Smith et al. (2017)	0.431	0.607	0.651
Conneau et al. (2017) - WaCky	0.451	0.607	0.651
Conneau et al. (2017) - fastText	**0.662**	**0.804**	**0.834**
Proposed method - fastText	0.377	0.565	0.625
Proposed method - WaCky	0.220	0.333	0.373

Table 1. *Comparing English-Italian results on Dinu's data.*

Ita-Eng	@1	@5	@10
Mikolov et al.	0.249	0.410	0.474
Faruqui et al.	0.310	0.499	0.570
Dinu et al.	0.246	0.454	0.541
Smith et al. (2017)	0.380	0.585	0.636
Conneau et al. (2017) - WaCky	0.383	0.578	0.628
Conneau et al. (2017) - fastText	**0.587**	**0.765**	**0.809**
Proposed method - fastText	0.310	0.502	0.547
Proposed method - WaCky	0.103	0.163	0.190

Table 2. *Comparing Italian-English results on Dinu's data.*

4.2.2. Experiments with the PanLex data

Using the PanLex database some experiments were made with different training set sizes. 3k training examples proved to be the best as Table 3 shows.

	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

Table 3. *Experiments with different training set sizes*

4.2.3. 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 4 summarizes the number of word pairs in the old and the new test sets. It should be noted that by this reduction principally the most 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 5 and on the PanLex data in Table 6. 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.

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

Table 4. *Word reduction of the new test sets*

4.2.4. 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 7 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.

4.2.5. 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 matrix, both the English-Italian and the English-Spanish dictionaries are used, according to Equation (2). 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 8. 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 9. 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.

5. 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.

	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 5. *Comparing Dinu's and PanLex data on Dinu's 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**

Table 6. *Comparing Dinu's and PanLex data on the PanLex test set*

	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

Table 7. *Continuing the baseline system with the PanLex data.*

	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 8. *Results of bilingual models trained pairwise on the three different languages.*

	L1-L2			L2-L1		
Precision	@1	@5	@10	@1	@5	@10
eng-ita	0.1573	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

Table 9. *Bilingual results of the multilingual model trained using three different languages at the same time.*

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 [8], than using Dinu's WaCky embeddings [1].

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. [1] 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.

6. Acknowledgements

This work is a collaboration of the Universitat Politècnica de València (UPV) and the Budapest University of Technology and Economics (BUTE). Work partially supported by the Spanish MINECO and FEDER founds under project TIN2017-85854-C4-2-R.

7. References

[1] G. Dinu, A. Lazaridou, and M. Baroni, "Improving zero-shot learning by mitigating the hubness problem," *arXiv preprint arXiv:1412.6568*, 2014.

[2] D. Kamholz, J. Pool, and S. M. Colowick, "Panlex: Building a resource for panlingual lexical translation." in *LREC*, 2014, pp. 3145–3150.

[3] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *arXiv preprint arXiv:1309.4168*, 2013.

[4] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 462–471.

[5] H. Youn, L. Sutton, E. Smith, C. Moore, J. F. Wilkins, I. Maddieson, W. Croft, and T. Bhattacharya, "On the universal structure of human lexical semantics," *Proceedings of the National Academy of Sciences*, vol. 113, no. 7, pp. 1766–1771, 2016.

[6] S. Ruder, I. Vulić, and A. Søgaard, "A survey of cross-lingual word embedding models," *arXiv preprint arXiv:1706.04902*, 2017.

[7] S. L. Smith, D. H. Turban, S. Hamblin, and N. Y. Hammerla, "Offline bilingual word vectors, orthogonal transformations and the inverted softmax," *arXiv preprint arXiv:1702.03859 (publised at ICRL2017)*, 2017.

[8] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," *arXiv preprint arXiv:1710.04087*, 2017.

[9] C. Xing, D. Wang, C. Liu, and Y. Lin, "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: Human Language Technologies*, 2015, pp. 1006–1011.

[10] A. Lazaridou, G. Dinu, and M. Baroni, "Hubness and pollution: Delving into cross-space mapping for zero-shot learning," in *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)*, vol. 1, 2015, pp. 270–280.

[11] W. Ammar, G. Mulcaire, Y. Tsvetkov, G. Lample, C. Dyer, and N. A. Smith, "Massively multilingual word embeddings," *arXiv preprint arXiv:1602.01925*, 2016.

[12] M. Artetxe, G. Labaka, and E. Agirre, "Learning principled bilingual mappings of word embeddings while preserving monolingual invariance," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2289–2294.

[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[15] J. Tiedemann, "Parallel data, tools and interfaces in opus." in *LREC*, 2012, pp. 2214–2218.