A Locally Linear Procedure for Word Translation

Soham Dan
University of Pennsylvania
sohamdan@seas.upenn.edu

Hagai Taitelbaum
Google Research
hagait@google.com

Jacob Goldberger
Bar-Ilan University
jacob.goldberger@biu.ac.il

Abstract

Learning a mapping between word embeddings of two languages given a dictionary is an important problem with several applications. A common mapping approach is using an orthogonal matrix. The Orthogonal Procrustes Analysis (PA) algorithm can be applied to find the optimal orthogonal matrix. This solution restricts the expressiveness of the translation model which may result in sub-optimal translations. We propose a natural extension of the PA algorithm that uses multiple orthogonal translation matrices to model the mapping and derive an algorithm to learn these multiple matrices. We achieve better performance in a bilingual word translation task and a cross lingual word similarity task compared to the single matrix baseline. We also show how multiple matrices can model multiple senses of a word.

1 Introduction

Continuous word embeddings are a standard component in many NLP applications. The embedding spaces can exhibit similar structures across languages. Several studies (Mikolov et al., 2013; Klementiev et al., 2012) have exploited this similarity by learning a linear mapping from a source to a target embedding space, and demonstrated this approach on word translation tasks. Xing et al. (2015) showed that using orthogonal matrices can significantly improve performance. Since then, several studies have aimed at improving these bilingual word embeddings by using bilingual word dictionaries generated in either supervised or unsupervised manner.

A multi-sense word in the source language can be translated into several different words in the target language. Studies have shown that for the multi-lingual translation problem, enforcing the transformation to be strictly orthogonal is too restrictive and performance can be improved by relaxing this constraint. It was shown that using orthogonalization as regularization rather than a strict constraint (Chen and Cardie, 2018), yields matrices that are close to orthogonal but are not necessarily exactly orthogonal, and improves performance (Taitelbaum et al., 2019a; Taitelbaum et al., 2019b). A single matrix however, whether orthogonal or not, cannot model the case of translating a multi-sense word.

In this study, motivated by the multi-sense word situation, we investigated another way to relax the common modeling assumption that a single orthogonal transformation is suitable for translating all the words of a source language into a target language. In our approach, the word translation procedure is modeled by a set of orthogonal matrices where each word is translated by one of the matrices. The words are grouped according to their associated matrix. This grouping is determined by applying a clustering procedure on the vocabulary words. Next, for each cluster we learn a separate orthogonal matrix. The word mapping function is thus modeled by a local orthogonal transformation. When translating a new word, we select the most suitable matrix for this specific translation task. We illustrate our approach on several standard word translation tasks and we show performance improvement compared to translation based on a single matrix.

Proceedings of the 28th International Conference on Computational Linguistics, pages 6013–6018
Barcelona, Spain (Online), December 8-13, 2020

2 Multi-Matrix Translation

Suppose we are given d dimensional word embeddings of a source language S and a target language T and a training dictionary of corresponding S-T word pairs. Denote this embedding dictionary by $(X, Y) = \{(x_t, y_t)\}_{t=1}^{n}$. The objective is to learn a mapping Q from the source space to the target space, such that, a source word is mapped to a vector that is close to its corresponding word in the target language.

Previous studies (Mikolov et al., 2013; Klementiev et al., 2012) have found that there is a linear correlation between the vector spaces of the two languages, thus, it is best to use a linear mapping. Xing et al. (2015) and Artetxe et al. (2016) showed that restricting the linear mapping Q to be orthogonal improves performance. The optimal mapping is found by minimizing the following cost function (F denotes the Frobenius norm):

$$\hat{Q} = \arg\min_{Q} \|QX - Y\|_F^2. \tag{1}$$

The solution to Eq. (1) is well-established and is known as the Procrustes Analysis (PA) algorithm (Schönemann, 1966). The optimal orthogonal mappings is obtained by $\hat{Q} = UV^\top$ where $U\Sigma V^\top$ is the Singular Value Decomposition (SVD) of $M = YX^\top$. This method has been used in many recent cross-lingual studies (Xing et al., 2015; Artetxe et al., 2016; Artetxe et al., 2017a; Conneau et al., 2017). However, it was shown that using one orthogonal matrix to map a vector space into another one can be too restrictive in some cases (Taitelbaum et al., 2019a).

We hypothesized that using multiple matrices would improve translation by relaxing the restrictive assumption of the ordinary PA solution (single matrix). The goal is to model the word translation by a set of matrices $Q_1, ..., Q_k$. The proposed algorithm is an extension of the single-matrix PA algorithm that was described above where the optimization goal remains minimizing the mean square error. The objective function we want to minimize is thus:

$$S(Q_1, ..., Q_k) = \sum_{t=1}^{n} \min_{i=1}^{k} \|Q_i x_t - y_t\|^2. \tag{2}$$

By minimizing the score (2) the aim is to simultaneously cluster the word pairs into k groups and assign the optimal translation matrix to each group separately. There is no closed form solution for this minimization problem. Instead we use an iterative procedure that resembles the k-means algorithm. Each iteration consists of two stages. In the first step we reassign each word pair to one of the clusters:

$$c_t = \arg\min_{i=1}^{k} \|Q_i x_t - y_t\|^2. \tag{3}$$

In the second step we re-estimate the translation matrices. The updated matrix Q_i is obtained by minimizing the following score.

$$S(Q_i) = \sum_{t|c_t=i} \|Q_i x_t - y_t\|^2. \tag{4}$$

The minimization of Eq. (4) can be efficiently computed by the PA algorithm. The alternate minimization procedure guarantees a monotone improvement of the score (2) until convergence to a local minima.

For alternate minimization algorithms, a good initialization is crucial for fast convergence. In our case it is reasonable to assume that two words whose embedding vectors are close should be translated by the same matrix. Thus, in order to find the initial cluster we can apply standard a k-means procedure on a transformation of the word embedding vectors. We adapt a dense transformation of the word vectors based on an auto-encoder representation. The autoencoder [1] is trained to minimize the reconstruction loss over the concatenation of source and target word embedding pairs. Then, k-means is applied on the latent representation of the word embedding vectors. Note that the k-means algorithm is also sensitive to the initialization. However, this problem is well-studied and there are good stable algorithms for k-means optimization such as k-means++ (Arthur and Vassilvitskii, 2007). Our training algorithm is summarized in Algorithm 1.

[1] The model dimension is $600 \times 100 \times 30 \times 100 \times 600$ neurons. The k-means step uses the 30-dimensional representation.

Algorithm 1 Multi-Matrix Translation Algorithm.

Required: Aligned set of dictionary word embeddings $(x_t, y_t),\ \ t = 1, ..., n.$
Task: Cluster the dictionary word pairs into k clusters.
Output: k mapping matrices $\{Q_i,\ \ i = 1, ..., k\}$.

while *Not converged* **do**

- Reassign each word pair to one of the clusters:

$$c_t = \arg\min_{i=1}^{k} \|Q_i x_t - y_t\|^2, \qquad t = 1, ..., n.$$

- Re-estimate the translation matrices by minimizing the following score:

$$S(Q_i) = \sum_{t|c_t=i} \|Q_i x_t - y_t\|^2, \qquad i = 1, ..., k.$$

end

Inference: The translation of a source word x to the target language is:

$$\hat{y} = \arg\max_{y \in V}(\max_i sim(Q_i x, y)$$

In the case where the translation is done by a single matrix Q, the standard inference procedure is the following. The translation of a source word embedding x to the target language is obtained by:

$$\hat{y} = \arg\max_{y \in V} sim(Qx, y) \tag{5}$$

where V is the vocabulary of the target language. $sim(x, y)$ can be, for example, the cosine similarity. However, in our case, there are k matrices which produce a different k word translations and we need to select the most suitable one. A natural solution is to use the translation similarity measure to decide on the correct matrix:

$$\hat{y} = \arg\max_{y \in V}(\max_i sim(Q_i x, y)). \tag{6}$$

We dub this inference process *Max-Scoring* inference. We dub the entire proposed procedure Multi-Matrix Model (MMM).

3 Experiments

Experimental Setup. In this experiment we empirically tested the advantages of using multiple matrices for the word translation task. We used the publicly available MUSE dictionaries (Lample et al., 2018)[2] for supervised mapping learning and evaluation. We used 300-dimensional pre-trained fastText word vectors, trained on Wikipedia (Bojanowski et al., 2017)[3]. The vectors were normalized to unit length and then zero centered (Artetxe et al., 2016). We conducted several experiments, all with English (*En*) as either the source or the target language. We report word translation results for *De, Es, Fr, It, Pt* in both directions.
Compared methods. We compared the following translation methods in our experiments:
(1) PA (Procrustes Analysis). The standard PA solution with one mapping from the source to the target space. We applied the same training procedure as in the supervised version of (Lample et al., 2018) and (Xing et al., 2015). Inference was done according to Eq. (5).

[2]https://github.com/facebookresearch/MUSE
[3]https://github.com/facebookresearch/fastText

	En-De	De-En	En-Es	Es-En	En-Fr	Fr-En	En-It	It-En	En-Pt	Pt-En	Avg.
PA	71.27	68.87	78.80	77.87	77.07	77.33	**73.33**	73.33	73.87	74.27	74.60
MMM	**72.13**	**69.73**	**79.2**	**78.47**	**78.2**	**78.67**	73.13	**75.67**	**74**	**75.4**	**75.46**

Table 1: Word translation accuracies (precision@1) for PA and MMM using the cosine similarity metric for inference, averaged over 5 runs. Results in bold are the best for each pair.

	En-De	De-En	En-Es	Es-En	En-Fr	Fr-En	En-It	It-En	En-Pt	Pt-En	Avg.
PA	**74.6**	**72.87**	81.33	82.6	81.27	82.53	76.6	77.07	79.67	79.53	78.81
MMM	74.53	72.33	**81.67**	**82.8**	**81.87**	**83.13**	**76.67**	**78.47**	**80**	**79.6**	**79.11**

Table 2: Word translation accuracies (precision@1) for PA and MMM using the CSLS metric for inference, averaged over 5 runs. Results in bold are the best for each pair.

(2) MMM (Multi-Matrix Model). An extension of the PA algorithm, with 2 orthogonal matrices. For $k \geq 3$ the gain in accuracy was not significant although, the MSE objective (2) still went down. Inference was done according to Eq. (6).

We report results for both cosine and Cross-domain Similarity Local Scaling (CSLS) (Lample et al., 2018) similarity metrics as the sim function in Eq. (5) and (6).

Results and Analysis. Tables 1 and 2 list the word translation results in terms of precision@1. We see that in most cases our Multi-Matrix model outperformed PA, indicating that using more than a single mapping from the source to the target space benefits in word translation task. The benefit is larger for the cosine similarity metric, which is expected since, our alternate minimization procedure uses Euclidean distance.

In Table 3 we present several examples where our proposed algorithm predicted a correct translation whereas the baseline method (PA) did not.

Pair	S	$k = 1$	$k = 2$
En-De	fewer	meisten	weniger
En-De	joke	peinlich	scherz
En-Fr	pushing	avançant	pousser
Fr-En	mesurer	measured	measuring
Fr-En	piscine	playground	pool

Table 3: Instances from our experiment where the single matrix method ($k = 1$) predicted a wrong translation for the source word (S) whereas our proposed method ($k = 2$) predicted a correct one.

To better understand the advantage of using two clusters, we analyzed the best scoring output from each cluster. We observed that multiple clusters captured different possible translations (and senses) of the same source word, as claimed in Sec. 1. For a fair comparison we also look at the top-k translations obtain by the PA solution, where k is the number of mapping matrices.

For example, in *Fr-En*, *measuring* and *measurement* are both correct translations of *mesurer* (*Fr*), each obtained by a different mapping. However, the top-2 predictions using the single matrix method are *measured* and *quantify* which are both incorrect. Another example is when translating *cumprimento* from Portuguese to English. Both *compliance* and *fulfillment*, which were produced by MMM, are correct translations, whereas the top-2 translations of PA were *compliance* (correct) and *obligations* (incorrect). Similarly, translating *maßstab* from German to English with MMM resulted in *yardstick* and *scale* which are both correct translations, whereas the top-2 translations of PA method were *scale* (correct) and *reproducible* (incorrect). In all of these cases, having two separate clusters produced the two senses. This demonstrates that by using two clusters we can better capture multiple senses of the source word. This phenomenon was observed across all language pairs.

Another interesting observation for the *En-Fr* pair involves gender differences. In French, there are different masculine and feminine translations for a word. The word *colombian* in English has separate entries in French for *colombian - colombien* (masculine) and *colombian - colombienne* (feminine). A single matrix predicted *colombienne* whereas using two matrices predicted *colombien* and *colombienne*.

Further, we verified that the second best translation in the single matrix case was incorrect (*vénézuélienne*). Another example is when translating *Prussian* (En) to French. The correct translations we obtained were *Prussien* (masculine) and *Prussienne* (feminine). Whereas the top-2 predictions by the PA algorithm were *Prussienne* and *Prusse* (meaning Prussia rather than Prussian). Each cluster in the two-matrix case generated one of the correct translations.

pair	NS	$k = 1$	$k = 2$
En-Es	0.63	0.72	0.73
En-De	0.60	0.72	0.73
En-It	0.65	0.73	0.73
Es-It	0.60	0.75	0.75
De-Es	0.55	0.72	0.72
De-It	0.56	0.70	0.71
Avg	0.60	0.72	0.73

Table 4: Results of the cross-lingual word similarity task. NS denotes the NASARI baseline.

We further evaluated the quality of our translation matrices using a word similarity task. We used the SemEval 2017 data (Camacho-Collados et al., 2017) which has pairs of nominals from the source and target languages respectively that are manually scored on similarity on a scale between 1-5. We translated each source embedding vector into the target embedding space using a system of k matrices as described above and measured how well it was correlated with the human labeled score using the Spearman Rho correlation coefficient. We also report the results of a baseline system, NASARI (Camacho-Collados et al., 2016). Note that the best performing systems in the competition used large knowledge bases such as ConceptNet whereas we only relied on bilingual dictionaries. In Table 4, we show that we obtained decent performance on this task by leveraging the bilingual dictionaries and when using multiple matrices we outperformed the single matrix results in (Lample et al., 2018). Using more than two matrices does not yield much improvement in the results.

4 Conclusion

In this paper, we presented a natural extension of the Procrustes Analysis algorithm that relaxes the current single-matrix modeling assumption. We empirically demonstrated the superiority of our method on two standard word translation tasks. Our proposed method based on multiple matrices can also be applied as a natural extension to several other related settings. In the future, we plan to investigate an unsupervised dictionary setting using our algorithm in the refinement steps (Artetxe et al., 2017b; Lample et al., 2018).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017a. Learning bilingual word embeddings with (almost) no bilingual data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017b. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26.

Xilun Chen and Claire Cardie. 2018. Unsupervised multilingual word embeddings. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *COLING*.

Guillaume Lample, Alexis Conneau, Marc-Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representation (ICLR)*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

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

Hagai Taitelbaum, Gal Chechik, and Jacob Goldberger. 2019a. A multi-pairwise extension of procrustes analysis for multilingual word translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hagai Taitelbaum, Gal Chechik, and Jacob Goldberger. 2019b. Multilingual word translation using auxiliary languages. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.