Translation with word embeddings

Nathan Bigaud
Roxane Cohen
Chloé Vildé

Master IASD

November 17, 2021

Contents

1 Introduction

1.1 Problem definition

In this paper we try to build a linear mapping between two latent spaces representing languages. X and Y are two latent spaces of vectors of size 300, X representing french monolingual embeddings, Y english monolingual embeddings. We have access to a dictionnary linking every word in X to its translation in Y. The goal is to build a translation matrix W such that $WX \approx Y$

We apply both supervised and a non-supervised approaches. Our main guides for the supervised approach were Mikolov, Le, and Sutskever 2013 and Xing et al. 2015, while we used Conneau et al. 2017b for the unsupervised approach.

1.2 Data and training strategy

We worked on the English and French MUSE dataset, using monolingual embeddings for French and English and the corresponding bilingual dictionnary.

Preprocessing was done by reducing the embeddings to those available in the dictionnary and reducing the many-to-many mapping to a one-to-one mapping by selecting the single most frequent translation of words with several possible translations. This shrinks the data use from two 200 000 embeddings to about 80 000 embeddings.

We trained our models on two datasets : a small sample of the 5500 most frequent words and the full dataset, in order to see if our methods are scalable.

2 Supervised Approaches

We here examine three approaches to the supervised problem, and demonstrate that while the closed form solution provides the best trade-off in terms of training time and results, the algorithm proposed by Xing et al. 2015 achieves better performance on the small dataset.

2.1 Motivation and setting

The supervised approach provides a solid baseline, as the learning problem it solves is simpler and simple approaches provide reliable results.

We examine three different formulations of the problem :

1. **Orthonormal W and $\ell 2$-norm**. The older form of the problem, known as the Procrustes Orthogonal problem. It has a known closed-form solution that we use

as baseline, as described by Schonemann 1966:

$$\begin{aligned}
&\min_{W \in O_d(\mathbb{R})} \|WX - Y\|_{\mathcal{F}} \\
&W^{\star} = \operatorname*{argmin}_{W \in O_d(\mathbb{R})} \|WX - Y\|_{\mathcal{F}} = UV^T \\
&\text{with } U\Sigma V^T = \text{SVD}\left(YX^T\right)
\end{aligned}$$

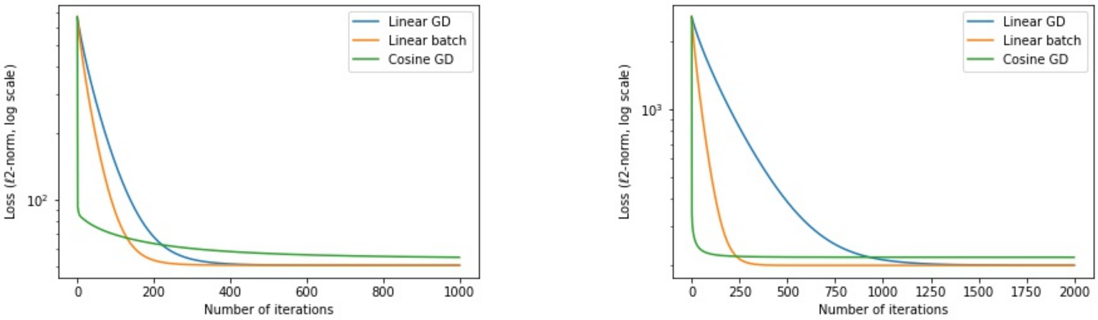
2. **Non-normalized W and $\ell 2$-norm**. The approach taken by the first paper we studied: Mikolov, Le, and Sutskever 2013:

$$\min_{W \in \mathbb{R}^d} \|WX - Y\|_{\mathcal{F}}$$

3. **Orthonormal W and cosine distance**. The latest supervised approach we explored in Xing et al. 2015:

$$\max_{W \in \mathbb{O}^d} (WX)^T Y$$

2.2 Training



(a) Small dataset of 5 500 most common words

(b) Large dataset of more than 80 000 words

Figure 1: Loss curves (using the $\ell 2$-norm) over 2000 iterations

The closed form solution computes in less than a few seconds, with minimal increase in training time with the size of the dataset : a 15-fold increase in the size of the dataset only increases computing time 7-fold. In in that aspect it by far outperforms gradient approaches.

The linear approach returns the expected results with minimal hyperparameter tuning. The classical gradient descent approach of the linear problem is the slowest to converge, while a batch approach achieves faster convergence (fig. 1)

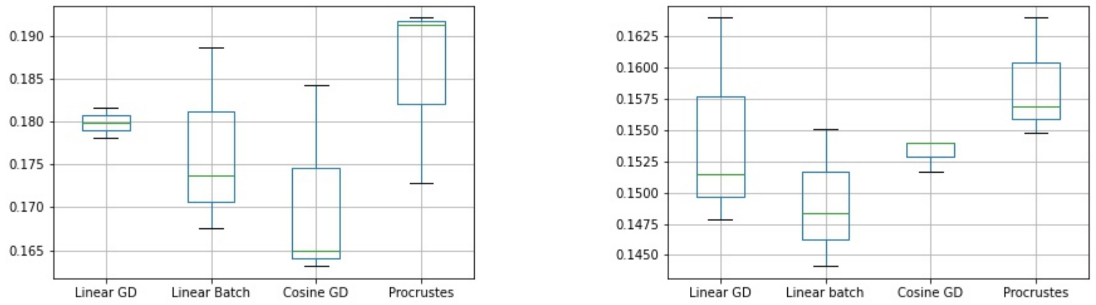
The cosine approach, after a first rapid drop of the $\ell 2$-norm at the first normalization step, converges faster to and as seen in the next section achieves better results on the small dataset. It converges to a higher absolute value than the linear approach, which is explained by the limited comparability of the loss' absolute value between a normalized and a non normalized approach. (fig. 1)

2.3 Results

We observe a clear gain in performance from the cosine method on the small dataset, on both top 1 error (fig. 2) and top 5 error (fig. 3). As explained in the paper, this is mostly due to the fact that the algorithm is trained using the same metric it is then evaluated on in the testing phase.

The cosine method does not outperforms other methods when training with our chosen hyperparameters. This does not reflect the expected result, and can be explained by a lack of hyper parameter tuning - given the limited training time and resources available.

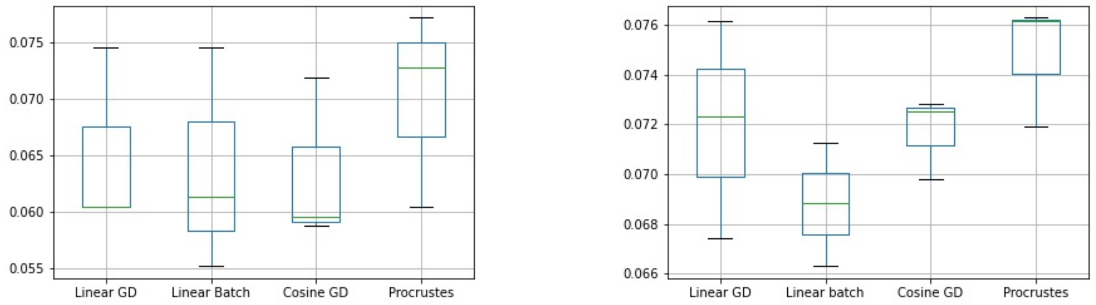
The closed form solution here doesn't generalize as well to new data. Comparing the training errors on the small dataset of the Procrustes and linear gradient approach, this appears to be due to overfitting.



(a) Small dataset of 5 500 most common words

(b) Large dataset of more than 80 000 words

Figure 2: Testing Top 1 error after training over 2000 iterations



(a) Small dataset of 5 500 most common words

(b) Large dataset of more than 80 000 words

Figure 3: Testing Top 5 error after training over 2000 iterations

3 Unsupervised Approach

From now on, we switch to the unsupervised approach presented in Conneau et al. 2017a. Briefly, we present how the algorithm works, according to fig. 4.

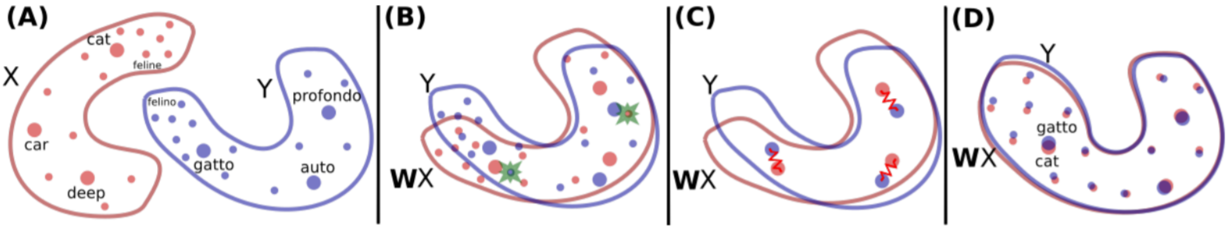


Figure 4: The unsupervised process presented in Conneau et al. 2017a

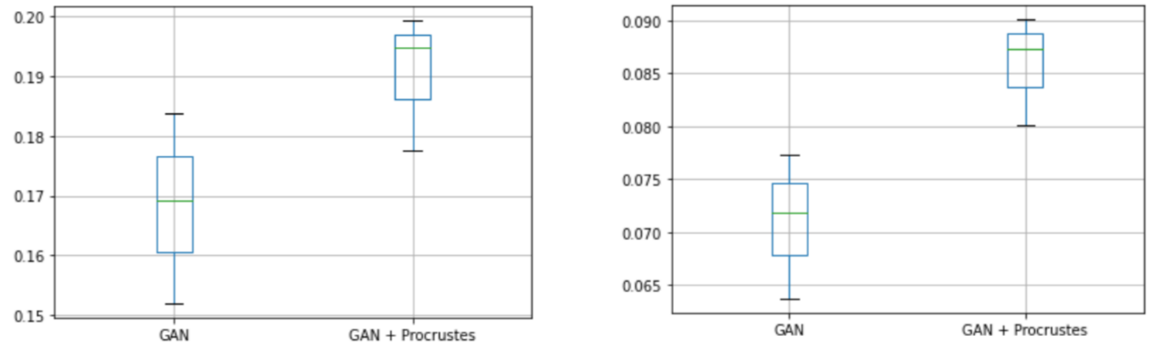
Recall that we want to translate a word from a language to another, by aligning the latent spaces X and Y. To do so, we use a GAN. The GAN's discriminator will try to distinguish true english embeddings from french embeddings that were translated into english. On the contrary, the GAN's generator is the matrix W, that should be the best possible linear mapping. We can perfect the alignement with the use of Procrustes problem or with a specific metric, the CSLS.

In this work, we implemented the three steps, available in the corresponding unsupervised notebook. GAN (step B) and GAN + Procrustes (step C) can be applied in a reasonnable amount of time for both datasets. However, our tentative implementation of the CSLS metric (without considering the Faiss algorithm, written in $C + +$) is not scalable and cannot be applied even to the small dataset in reasonable time. For this reason, we only provide results for step B and step C.

Our methodology is the same as before : once we have the two datasets, we split them into train-test sets and learn the mapping matrix W using either GAN or GAN + Procrustes on the training set, before computing results on the test set. This process is repeated 3 times. For the GAN, we use the same hyperparameters as in Conneau et al. 2017a

Results for the small dataset are displayed in fig. 5. A longer training could give better results, on the small and full datasets. We observe quite good performances for the GAN only. Notice these **results are robust under several train-test split**. Moreover, it only takes 5 minutes to obtain averaged results. Surprisingly, **GAN achieves better performances than GAN + Procrustes on the small dataset**. Indeed, Procrustes on the small dataset is less relevant as the small dataset already keeps the most important words.

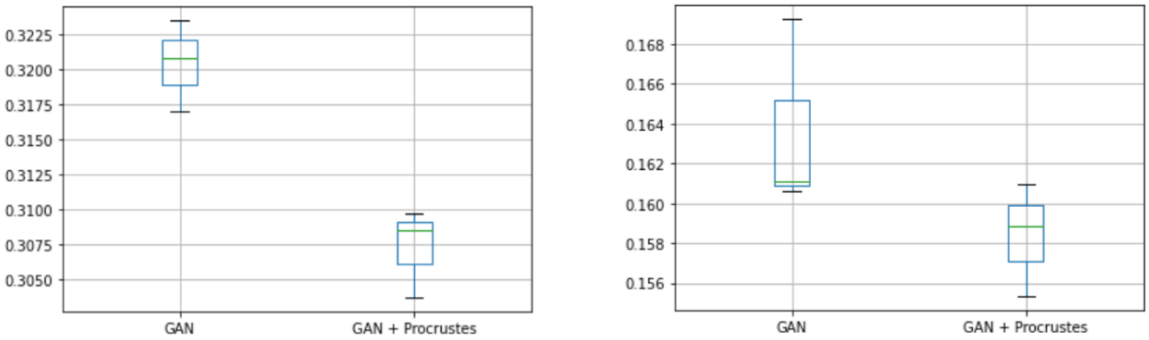
Concerning the full dataset, results are given in 6, we obtain higher error rates with GAN only, with smaller variance. This time, training is longer as applying procrustes on the full dataset is computationally intensive. However, we observe that **GAN + Procrustes performs better than a single GAN**. This was expected, as on the full dataset, the disparity of words frequency is higher.



(a) Top 1 error on the small dataset

(b) Top 5 error on the small dataset

Figure 5: Error rate on the small dataset



(a) Top 1 error on the full dataset

(b) Top 5 error on the full dataset

Figure 6: Error rate on the full dataset

4 Perspectives and conclusion

To sum-up, we obtained strong and reliable results with both supervised and unsupervised methods. However, we would have planned to experiment perspectives, with more time.

Indeed, in the supervised case, we used heuristics and small scale testing only to define learning rates and we could extend the hyperparameters tuning. Similarly, the GAN hyperparameters were set according to Conneau et al. 2017a as a proper hyperparameter tuning was not scalable to the full dataset.

Moreover, we can consider the robustness to changes in data of our models. We here work in a simplified setting where a bilingual dictionnary is available and we have selected the data to be a one to one matching problem. We would here explore how taking partially matching dictionnaries of different sizes $n \times 300$ and $m \times 300$, and shrinking them to the same size using linear dimentionality reduction would influence performance.

Finally, one should consider that word-to-word translation is not the best suited method for translation. Nowadays, RNNs, LSTMs and gated RNNs are the most popular methods for language translation. However, they handle sequences word-by-word which leads to an obstacle toward parallelisation of the process. Furthermore, when these sequences are too long, the model forgets the content of distant positions in sequence or mix it with following positions' content.

In Vaswani et al. 2017, authors proposed a new model architecture, transformers, which relies on an attention mechanism to draw global dependencies between input and output. This solution allows for significantly more parallelisation and tremendously improves translation quality.

References

Conneau, Alexis et al. (2017a). "Word Translation Without Parallel Data". In: *CoRR* abs/1710.04087. arXiv: `1710.04087`. URL: `http://arxiv.org/abs/1710.04087`.

– (2017b). "Word translation without parallel data". In: *arXiv preprint arXiv:1710.04087*.

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

Schonemann, Peter H (1966). "A generalized solution of the orthogonal procrustes problem". In: *Psychometrika* 31.1, pp. 1–10.

Vaswani, Ashish et al. (2017). "Attention Is All You Need". In: *CoRR* abs/1706.03762. arXiv: `1706.03762`. URL: `http://arxiv.org/abs/1706.03762`.

Xing, Chao et al. (2015). "Normalized word embedding and orthogonal transform for bilingual word translation". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1006–1011.