Rethinking Wasserstein-Procrustes for Aligning Word Embeddings Across Languages

May 2020

Joint Bachelor Thesis for Degrees in

MATHEMATICS
and
ENGINEERING PHYSICS

Guillem Ramírez Santos

Under the kind guidance of

Prof. Marin Soljačić
and
Rumen Dangovski
at
Massachusetts Institute of Technology

Abstract

The emergence of unsupervised word embeddings, pre-trained on very large monolingual text corpora, is at the core of the ongoing neural revolution in Natural Language Processing (NLP). Initially introduced for English, such pre-trained word embeddings quickly emerged for a number of other languages. Subsequently, there have been a number of attempts to align the embedding spaces across languages in a supervised, semi-supervised, or an unsupervised manner, which could enable a number of cross-language NLP applications. In supervised approaches, the alignment is typically done according to the rotation that minimizes the Frobenius norm in the Procrustes problem, which has a closed-form solution, easily obtainable using singular value decomposition (SVD). The unsupervised formulation of the problem is more challenging as it needs to optimize the much harder Wasserstein-Procrustes objective, and thus, in practice, most approaches resort to some modification of this objective. In the present work, we demonstrate how properties of problems equivalent to Wasserstein-Procrustes can help in the unsupervised setup. We further show that, both in the supervised and unsupervised setups, strong baselines can be improved by a rather simple algorithm that optimizes the Wasserstein-Procrustes objective. Finally, a modification of this algorithm using just a little supervision can yield satisfactory results. We believe that our rethinking of the Wasserstein-Procrustes problem would enable further research and would eventually help develop better algorithms for aligning word embeddings across languages.

Keywords Bilingual Lexicon Induction, Word Embeddings, Natural Language Processing, Translation

MSC2020 68T50

Acknowledgements

First of all, I would like to thank my supervisor, Rumen Dangovski, for the countless hours he has spent discussing this or other pieces of research with me. I have learned a lot from our discussions and your guidance, and I appreciate that you were always trying to help me, no matter how full of work you were. I would also like to thank Preslav Nakov for his guidance, the discussions we have shared and his thoughtful corrections of this thesis. I am also very grateful to Marin Soljačić for the discussions that we had, and especially for giving me the opportunity of carrying out this research in his group.

Many thanks to the Centre de Formació Interdisciplinària Superior (CFIS) from the Universitat Politècnica de Catalunya (UPC) and Soljačić's group at the Massachusetts Institute of Technology (MIT) for funding my stay.

I would like to thank Xavier Giró for being the tutor of this thesis at Universitat Politècnica de Catalunya. I appreciate his support and that he has helped me to make this easier. This thesis has been written using a template from Andrea Calafell. Thanks to Martín Forsberg for his help with the cover page. Finally, thanks to Rong Rong Hu for her esthetic correction of the layout.

Contents

Chapter 1

Introduction

Pre-trained word embeddings, which map words to dense vectors of low dimensionality, have been the key enabler of the ongoing neural revolution, and today they serve as the basic building blocks of the vast majority of the contemporary Natural Language Processing (NLP) models. While initially introduced for English only (Chen et al. 2013, Pennington et al. 2014, Bojanowski et al. 2017, Joulin et al. 2017), pre-trained embeddings quickly emerged for a number of other languages, (Heinzerling & Strube 2018) and soon the idea of cross-language embedding spaces was born. In a cross-language embedding space, two semantically similar (or dissimilar) words would be close to (or far from) each other regardless of whether they are from the same or from different languages. Using such a space is attractive, as for a number of NLP tasks, it enables the application of an NLP model trained for one language to test input from another language. Ideally, such spaces could be trained on parallel bilingual datasets, but as such resources are of limited size (compared to large-scale monolingual resources typically used to pre-train monolingual word embeddings), it has been more attractive to train monolingual word embeddings for different languages independently, and then to try to align the corresponding embedding spaces in what is commonly known as bilingual lexicon induction. This has been attempted in a supervised (Le et al. 2013, Irvine & Callison-Burch 2013) semi-supervised (Artetxe et al. (2017) used as little supervision as a 25 word dictionary or even an automatically generated list of numerals) or an unsupervised (Lample et al. 2017) manner.

Initial attempts at aligning the spaces used a dictionary of cross-language word translation pairs as anchors between the two spaces in order to infer the nature of the transformation that relates the first language to the second one (Le et al. 2013). This is a supervised setup, where the alignment is typically done according to a rotation that minimizes the Frobenius norm in the Procrustes problem, which has a closed-form solution, easily obtainable via SVD.

Procrustes. Given two ordered clouds of points $X, Y \in \mathbb{R}^{N \times d}$, each with N points of dimension d, the orthogonal Procrustes problem finds the orthogonal matrix $W \in \mathbb{R}^{d \times d}$ that minimizes the following Frobenius norm:

$$\underset{W \in \mathbb{R}^{d \times d}}{\arg\min} \|XW - Y\|_2^2 \tag{1.1}$$

For the translation of word embeddings, W is taken to be an orthogonal matrix due to a self-similarity argument (Smith et al. 2017): if W is the orthogonal transformation matrix, then its transpose W^T is the inverse map, and any word embedding x can be recovered by $W^T W x$. Besides, the cosine similarity between a source word x and a target word y would be independent of the space where this similarity is measured: $y^T W x / \|Wx\| \|y\| = x^T W^T y / \|x\| \left\|W^T y\right\|$. The convenience of using an orthogonal matrix have also been supported empirically (Xing et al. 2015, Zhang et al. 2016, Artetxe et al. 2016). The orthogonal Procrustes problem has a closed-form solution $W = UV^T$, where $U\Sigma V^T$ is the singular value decomposition (SVD) of $X^T Y$ as shown by Schönemann (1966).

A popular unsupervised formulation of the problem is known as the Wasserstein-Procrustes (Hoshen & Wolf 2018, Alaux et al. 2019), which is more challenging as it needs to optimize a gen-

eralization of the Procrustes objective. In practice, most approaches resort to some modification of this objective.

Wasserstein-Procrustes. Given two clouds of points X, $Y \in \mathbb{R}^{N \times d}$, each with N points of dimension d, the Wasserstein-Procrustes problem finds the orthogonal matrix $W \in \mathbb{R}^{d \times d}$ and the permutation matrix $P \in \mathbb{R}^{N \times N}$ that minimize the Frobenius norm:

$$\underset{P \in \pi(N), W \in O(d)}{\arg\min} \|XW - PY\|_2^2 \tag{1.2}$$

where $\pi(N)$ are the N-dimensional permutations matrices and $O(d)$ are the d-dimensional orthogonal matrices.

This problem has recently gained considerable interest for two main reasons. First, a bilingual dictionary is not always available, and even when available, it might not be good enough to obtain high-quality mapping between the two spaces. Second, there are a number of semi-supervised and even unsupervised approaches that claim better results than the supervised approaches (Lample et al. 2018, Grave et al. 2019, Alvarez-Melis & Jaakkola 2018, Hoshen & Wolf 2018). Although these methods are inspired by Wasserstein-Procrustes, they deviate from Objective 1.2 and solve a modified problem, as described below.

- GANs optimisation was first introduced for bilingual lexicon induction by Barone & Valerio (2016), but its cannonical implementation was done by Lample et al. (2018), which present *multilingual unsupervised and supervised embeddings* (MUSE): an adversarial method in which the transformation matrix W is considered as a generator, and thus is trained so that the mapped word embeddings XW cannot be distinguished from the set Y via a discriminator, by a generative adversarial net (Goodfellow et al. 2014).

  However, we note that MUSE's objective is not exactly that of Wasserstein-Procrustes, as we describe in the sketch on Figure 1.1. Suppose that languages X and Y have only two different two-dimensional words, where X consists of two words in an angle of 90 degrees, whereas Y has two words separated by 180 degrees. Then the rotation that minimises the Wasserstein-Procrustes objective function is not the same as the rotation that makes it more difficult to predict whether an element belongs to XW or to Y, as illustrated in Figure 1.1.

- Grave et al. (2019) suggest an iterative procedure whose initial condition minimizes the convex relaxation $\left\|X^T P Y\right\|_2^2$ instead of the original problem (which we will develop to Equation 2.8 in Chapter 2). This relaxation is known as the Gold-Rangarajan relaxation and can be solved using the Frank-Wolfe algorithm. The solution to this relaxation is then used as the initial condition for a gradient-based iterative procedure that stochastically samples different subsets of words for which there is not necessarily a direct translation. This deviates strongly from Objective 1.2: not only the initial condition does not optimise Wasserstein Procrustes, but also the iterative procedure does not optimise it as it translates words that are not necessarily the optimal match.

- Alvarez-Melis & Jaakkola (2018) use the concept of Gromov-Wasserstein distance to provide an alternative to Wasserstein-Procrustes. This distance does not operate on points but on pairs of points, turning the problem from a linear to a quadratic one. This new loss function can be optimized efficiently with first-order methods, whereby each iteration

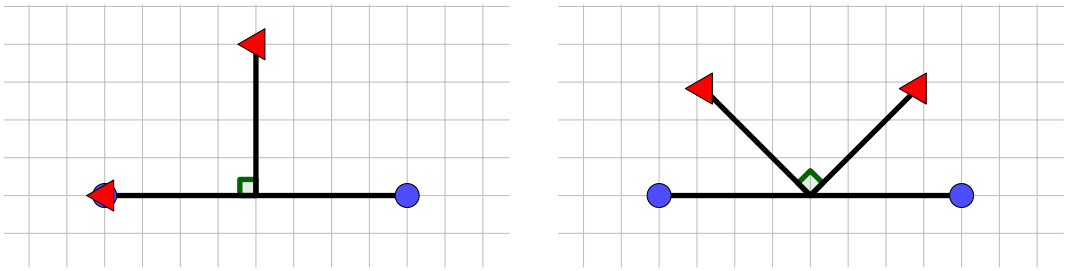

Figure 1.1: **GANs do not optimize Wasserstein-Procrustes.** Given two languages (red triangles, separated by 90 degrees, and blue circles, separated by 180 degrees) compound each of them by two different words in a two-dimensional space, the left shows the transformation an adversarial approach would find. Right shows the distribution that minimises the Frobenius norm in the Wasserstein-Procrustes problem.

involves solving a traditional optimal transport problem. Artetxe et al. (2018) achieve better results by combining this idea with a refinement method called stochastic dictionary induction, i.e., randomly dropping out dimensions of the similarity matrix when extracting a seed dictionary for the next iteration of Procrustes Analysis.

- Hoshen & Wolf (2018) are inspired in the Iterative Closest Point (ICP) method used in 3D point cloud alignment. Although their transformation matrix is not necessarily orthogonal, this property is enforced using a regularization. Their fundamental difference to Objective 1.2 is, however, that they minimise the norm L_1 rather than the L_2. This difference in the norm might have relevant consequences: L_2 norm tries to avoid higher values whereas L_1 is often used in feature selection in order to achieve a higher degree of sparsity. Hence, the L_1 norm tries to achieve a better map for certain words, whereas the L_2 norm might achieve a better overall map.

With the above consideration that current unsupervised methods do not optimize Wasserstein-Procrustes anymore, they nevertheless provide good accuracy for synthetically generated dictionary induction tasks. Therefore, here we ask the following questions: Can we find approximate solutions to the original Wasserstein-Procrustes Objective 1.2 that not only minimize the objective in general, but also provide good accuracy on dictionary induction tasks? Can we take existing methods and improve them further by using refinements that optimize Objective 1.2? Can we find natural scenarios for which we find good solutions?

Here, we discuss properties of the Problem 1.2 and propose an iterative algorithm inspired by them. We try this algorithm in different scenarios: (i) in an unsupervised map of data in the same language, (ii) in the case where we already know an orthogonal transformation W for a translation of two languages and want to refine it, and (iii) in the case where we can apply some supervision.

Our contributions can be summarized as follows:

1. We derive different natural initialization W_0 of the transformation matrix in the Wasserstein-Procrustes problem.
2. We propose an iterative algorithm that attempts to solve the Wasserstein-Procrustes problem exactly. We study under what circumstances that algorithm converges to the optimal

solution. We find that our algorithm can improve over strong baselines, when used as a refinement tool.

3. We validate empirically that our approach works well in realistic scenarios, where small-size supervision is present.

The rest of this thesis is organized as follows: In Chapter 2, we demonstrate key properties of the Wasserstein-Procrustes problem, which is at the core of our approach. In Chapter 3, we describe our algorithms. In Chapter 4, we discuss our experiments and results. In Chapter 5, we discuss related work. In Chapter 6, we conclude and we point to promising directions for future work.

Chapter 2

Properties of the Wasserstein-Procrustes Problem

We begin by simplifying Objective 1.2 to arrive at some essential properties.

Equivalent Formulation Problem 1.2 is equivalent to maximizing the trace norm on the permutation matrix $X^T PY$ over P, as we demonstrate below. First, we convert the norm to a Frobenius inner product $\langle A, B\rangle_F = \mathrm{Tr}\big(A^T B\big)$, as follows:

$$\begin{aligned}
&\underset{P\in\pi(N),W\in O(d)}{\arg\min} \|XW - PY\|_F^2 = \\
&\underset{P\in\pi(N),W\in O(d)}{\arg\min} \langle XW - PY, XW - PY\rangle_F = \\
&\underset{P\in\pi(N),W\in O(d)}{\arg\min} \langle XW, XW\rangle_F + \langle PY, PY\rangle_F \\
&\qquad\qquad - 2\langle XW, PY\rangle_F = \\
&\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle XW, PY\rangle_F,
\end{aligned}$$

where we used that $\|XW\|^2 = \|X\|^2$ and $\|PY\|^2 = \|Y\|^2$ since P and W are orthogonal. Since these two norms are independent of P and W, we can ignore them for the optimization.

Now, we use the cyclic property of the trace,

$$\mathrm{Tr}(ABC) = \mathrm{Tr}(CAB) = \mathrm{Tr}(BCA),$$

to obtain as follows:

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle XW, PY\rangle_F = \tag{2.1}$$

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \mathrm{Tr}\big(W^T X^T PY\big) = \tag{2.2}$$

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle W, X^T PY\rangle_F = \tag{2.3}$$

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle W, X^T PY\rangle_F = \tag{2.4}$$

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle W, U\Sigma V^T\rangle_F = \tag{2.5}$$

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle S, \Sigma\rangle_F \tag{2.6}$$

where

$$U(P)\Sigma V(P)^T = \mathsf{SVD}(X^T PY)$$

and $S = U^T W V$ for $U \equiv U(P)$ and $V \equiv V(P)$. Note that S is orthogonal since it is the product of orthogonal matrices, which implies it must be the identity $\mathbb{I}_d$ in order to maximize the Frobenius inner product. Therefore, we derive that

$$\underset{P \in \pi(N), W \in O(d)}{\arg\max} \langle S, \Sigma \rangle_F = \underset{P \in \pi(N)}{\arg\max} \operatorname{Tr}(\Sigma) \tag{2.7}$$

$$= \underset{P \in \pi(N)}{\arg\max} \left\| X^T P Y \right\|_* \tag{2.8}$$

where $\|\cdot\|_*$ denotes the nuclear norm. The condition on W fulfills that $U^T W V = \mathbb{I}_d$, where both $U(P)$ and $V(P)$ are taken at the optimum value P^*. Minimizing the nuclear norm is generally expensive and Grave et al. (2019) suggested replacing it by the Frobenius norm. However, here we continue to optimize the trace norm with the goal of not deviating from Objective 1.2.

Hungarian algorithm Given two clouds of points X, $Y \in \mathbb{R}^{N \times d}$, each with N points of dimension d, we find the permutation matrix P that gives the correspondence between the different points according to the following problem:

$$\underset{P \in \pi(N)}{\arg\min} \|X - PY\|_2^2 \tag{2.9}$$

repeating the last proof for the case where we do not want to optimize W, which is taken as the identity, we see that this problem is equivalent to the following linear program

$$\underset{P \in \pi(N)}{\arg\min} \operatorname{Tr}\left(X^T P Y\right), \tag{2.10}$$

which is the maximum weight matching problem. It can be solved through the Hungarian algorithm, which has a complexity of $O(n^3)$ (Tomizawa 1971).

Equivalent problems One useful property of the trace norm is that $\|UA\|_* = \|AV\|_* = \|A\|_*$, where U and V are orthogonal matrices. Knowing this, and calculating the SVD for both X and Y, we obtain the following

$$\underset{P \in \pi(N)}{\arg\max} \left\| X^T P Y \right\|_* = \underset{P \in \pi(N)}{\arg\max} \left\| V_X \Sigma_X U_X^T P U_Y \Sigma_Y V_X^T \right\|_*$$

which yields

$$\underset{P \in \pi(N)}{\arg\max} \left\| \Sigma_X U_X^T P U_Y \Sigma_Y \right\|_*. \tag{2.11}$$

Let us define $\widetilde{X} = U_X \Sigma_X$ and $\widetilde{Y} = U_Y \Sigma_Y$. Then, the optimal solution P would be the same for translations involving all of the following pairs of word embeddings: (X, Y), ($\widetilde{X}$, Y), (X, $\widetilde{Y}$) and ($\widetilde{X}$, $\widetilde{Y}$). However, the optimal transformation matrix W^* will be different for each of these problems. There is a different, yet interesting way of looking at this: if we follow the iterative procedure that starts from an initial transformation matrix $X_0 = XW_0$, and then we want to solve the problem (2.10), the equivalent problems will induce a set of *natural initializations* of the transformation W, which we formalize below.

Natural initialization: The iterative procedure

$$\begin{aligned} X_0 &= XW_0 \\ P_n &= \underset{P\in\pi(N)}{\arg\min}\,\mathrm{Tr}(X_n^T P Y_n) \\ Y_{n+1} &= P_n Y_n \\ W_n &= \underset{W\in\mathbb{R}^{N\times N}}{\arg\min}\,\|X_n W - Y_{n+1}\|_2^2 \end{aligned}$$

that tries to minimise Wasserstein-Procrustes aims for the same solution P as the problems with initial conditions

$$\begin{aligned} X_0 &= XV_X W_0 \\ X_0 &= XW_0 V_Y^T \\ X_0 &= XV_X W_0 V_Y^T. \end{aligned}$$

The significance of the natural initialization is that it gives us a starting point for a different problems that have the same solution P. It must be noted that these transformations of X_0 are not the unique ones that will have the same original solution, as the trace norm is invariant by any orthogonal transformation; however, they were useful in order to avoid bad local minima as it will be showed in Chapter 4.

We now utilize the insights from this chapter to proceed with describing our approach.

Chapter 3

Approach

In this chapter, we present a general iterative algorithm that attempts to solve the Wasserstein-Procrustes problem.

Joint optimization on W and P For the Wasserstein-Procrustes problem from Equation 1.2, a joint iterative procedure involving the Procrustes problem and the Hungarian algorithm has been dismissed due to its computational cost and convergence to bad local minima. However, as we will show below, there are a number of situations where such a simple approach can be extremely beneficial at the cost of very few improvements based on the previous chapter.

Algorithm 2 is the most general iterative procedure that we consider here, and it serves as the backbone for our experiments. In the next chapter we describe the utilization of our improvements in detail.

Variants of the natural initialization. The first of the improvements is considering the different equivalent problems or the natural initialization transformations we have seen from the previous chapter. We observe empirically that besides the four problems that share the same optimal P it is possible to improve results by considering the opposite optimization problem: instead of maximising the costs for the two clouds of points (X, Y), sometimes minimising the costs leads to a solution with a higher trace norm and eventually converges to a better solution. Minimization of costs is achieved by simply considering the cloud $-X$ instead of X.

Supervised translation. We also consider the event of supervised translation. There are many different ways of doing this, but the procedure that converges the fastest is to force n couples when calculating the Hungarian map, where typically $n \ll N$. We also consider similar approaches,

Algorithm 1 Cut Iterative Hungarian (CIH)

$X \leftarrow XW_0$
$\left\|X_{NEW}^T Y_{NEW}\right\|_* = \infty$
while $\left\|X^T PY\right\|_* < \left\|X_{NEW}^T Y_{NEW}\right\|_*$ **do**
  $\left\|X^T PY\right\|_* \leftarrow \left\|X_{NEW}^T Y_{NEW}\right\|_*$
  $X \leftarrow X_{NEW}$
  $Y \leftarrow Y_{NEW}$
  P $\leftarrow$ Hungarian (X, Y)
  W $\leftarrow$ Procrustes (X, PY)
  $X_{NEW} \leftarrow XW$
  $Y_{NEW} \leftarrow PY$
end while

Algorithm 2 Iterative Hungarian (IH)

$X \leftarrow XW_0$
$\left\|X_{NEW}^T Y_{NEW}\right\|_* = \infty$
while $\left\|X^T PY\right\|_* < \left\|X_{NEW}^T Y_{NEW}\right\|_*$ **do**
  $\left\|X^T PY\right\|_* \leftarrow \left\|X_{NEW}^T Y_{NEW}\right\|_*$
  $X \leftarrow X_{NEW}$
  $Y \leftarrow Y_{NEW}$
  for x, y **in** EquivalentProblems (X, Y) **do**
    $P \leftarrow$ Hungarian (x, y)
    if $\left\|x^T Py\right\|_*^2 < \left\|X_{NEW}^T Y_{NEW}\right\|_*$ **then**
      $W \leftarrow$ Procrustes (x, Py)
      $X_{NEW} \leftarrow xW$
      $Y_{NEW} \leftarrow Py$
      $\left\|X_{NEW}^T Y_{NEW}\right\|_* \leftarrow \left\|x^T Py\right\|_*^2$
    end if
  end for
end while

for instance deciding how to update Algorithm 2, taking into account the accuracy of the maps in a small subset of the data. With these improvements we show similar results to the stronger supervised ones at the cost of more iterations to converge. Choosing among these methods could be motivated by how trustful the maps from the initial dictionary are. By *trustful* here we consider how many of the corresponding cloud points are correctly matched.

We use a fast implementation of the Hungarian algorithm (https://github.com/cheind/py-lapsolver) for dense matrices based on shortest path augmentation (Edmonds & Karp 1972). Relaxations of the original problem can achieve higher speeds. Cuturi (2013) showed how smoothing the classical optimal transportation problem with an entropic regularization term results in a problem which can be computed through Sinkhorn-Knopp's matrix scaling algorithm at a speed that is several orders of magnitude faster than that of transportation solvers.

Mapping. Although our method provides a permutation matrix P, this has certain limitations: this is not necessarily the best possible map as languages translations are not a one-to-one map. Nearest neighbors has been typically used but it suffers from the so-called hubness problem: in high-dimensional vector spaces certain vectors are universal nearest neighbors (Radovanovic et al. 2010) and this is a common problem for word-embedding-based bilingual lexicon induction (Dinu & Baroni 2014). Lample et al. (2018) presented *cross-domain similarity local scaling* (CSLS), which is a method intended to reduce the influence of hubs by expand high-density areas and condense low-density ones. Given one source vector x, the mean similarity of its transformation Wx to its k nearest neighbors is defined as $\mu_S^k(Wx) = \frac{1}{k}\sum_{i=1}^k \cos(x, f_i)$, being $\cos$ the cosine similarity and f_i the i-th nearest target neighbor of Wx. Then $\mu_T^k(f_i)$ is defined analogously for every i, and $CSLS(x,\ f_i)$ is calculated as $2\cos(x, f_i) - \mu_S^k(Wx) - \mu_T^k(f_i)$. Intuitively, this mapping increases the similarity associated with isolated word vectors. Conversely it decreases the ones of vectors lying in dense areas. For the following experiments, we use the map induced by the nearest neighbor according to CSLS with $k = 10$.

Chapter 4

Experiments

In our first set of experiments we recreate the benchmarks that have been studied in Grave et al. (2019). These experiments correspond to mapping of word embeddings in English to another set of word embeddings trained in a similar fashion and in the same language. In the second set of experiments we propose a way in which our algorithm could be used for refinement purposes. Finally, we evaluate our algorithm in the framework of machine translation with little supervision.

4.1 Benchmarks from Grave et al. (2019)

The first set of experiments justify that the simple iterative procedure displayed in Algorithm 2 works and explain under what circumstances it can be relaxed or needs some help in the form of either supervision or a natural initialization matrix W_0. For the following controlled experiments the initialization matrix has been set to be the identity. We design the following four approaches:

- *Hungarian.* Running the Hungarian algorithm for only one iteration, and then taking the permutation matrix P as the map.
- *Cut Iterative Hungarian (CIH).* Running the Hungarian algorithm with the Algorithm 2 updates $X \leftarrow XW$ and $Y \leftarrow PY$ (see Algorithm 1).
- *Iterative Hungarian (IH).* Running the previous iterative procedure but considering the different natural initializations (see Algorithm 2).
- *Supervised Iterative Hungarian (SIH).* We learn 5% of the total mapped words coming from the supervision, and then we perform IH for the rest of the words.

The experiments from this section recreate the ones in Grave et al. (2019). We use FastText (Bojanowski et al. 2017, Joulin et al. 2017) to train word embeddings on 100M English tokens from the 2007 News Crawl corpus.[1] The different experiments in this section consist of changing the different training conditions and correctly mapping the results. Models are trained using Skipgram (Sutskever et al. 2013) unless stated otherwise, and using the standard parameters of FastText.[2] The experiments are described as follows:

- **Seed.** We only change the seed used to generate the word embeddings in our fastText runs. The source and the target are both word embeddings trained using the same parameters
- **Data.** We separate the dataset in two equal parts. We train corresponding word embeddings from the two separate parts. The source and the target correspond to word embeddings trained with the same parameters but trained on different data.

[1] http://statmt.org/wmt14/translation-task.html
[2] https://github.com/facebookresearch/fastText

- **Window.** We train the models with window sizes of 2 and 10 respectively. The source and the target correspond to word embeddings trained on the same data but with a different window size.
- **Algorithm.** We train the first algorithm with Skipgram and the second one with CBOW (Sutskever et al. 2013). The source and the target correspond to word embeddings trained on the same data but with a different method.

	Seed	Window	Algorithm	Data
Hungarian	**99%**	7%	7%	1%
CIH	**100%**	**100%**	**100%**	0%
IH	**100%**	**100%**	**100%**	0%
SIH	**100%**	**100%**	**100%**	**100%**

Table 4.1: Results for the first set of experiments.

We run the above algorithms on the 10,000 most frequent words. Table 4.1 shows the results for the different algorithms. The method used for doing the final map is the nearest neighbor for CSLS with $k = 10$. The percentage is taken over all the words from the model. We comment on the method and results below.

- The supervised approach seems to work well with very little supervision, but all other attempts fail when facing the problem of mapping data from different datasets. An interpretation of this is that by adding some supervision we are improving the initial W_0 and therefore this method starts from a better initial condition than others. This effect may be similar (although having less impact) than the help introduced in the IH with the equivalent problems or the natural initial transformations.
- The first three experiments converged in three or less iterations. SIH took around twenty iterations to converge for the data experiment.
- The Hungarian algorithm, which is not designed for the Wasserstein-Procrustes method, correctly finds the map for the seed experiment, whereas some other reported iterative experiments fail at achieving good results with this experiment (Grave et al. 2019).

The proposed iterative procedures converge, but a good minimum is achieved depending on the initial conditions and the help of supervision or equivalent problems. This suggests that the Algorithm 1 could work well as long as we start from an initial transformation matrix W_0 close enough to the true solution.

The importance of the initial condition can be showed by the natural initial conditions. The solution of the four different equivalent problems induce different optimal transformation matrices W^*. In the first iteration of IH, a branch among these four is chosen. Table 4.2 shows the Euclidean distance between each of the four natural initialization (assuming $W_0 = \mathbb{I}$) and their respective optimal solution W^* for the four experiments. These distances are different for the four branches, and being able to choose the best one (the one that minimises this distance) is key for the convergence of the algorithm.

Distances that are too big do not converge into a good solution. For the experiment of the seed, such a small distance justifies that a single iteration of the Hungarian algorithm was enough

	Seed	Window	Algorithm	Data
$\mathbb{I}$	**9.49**	**12.59**	**12.45**	**14.11**
V_X	14.13	14.14	14.18	14.19
V_Y^T	14.15	14.18	14.18	14.14
$V_X V_Y^T$	13.95	14.10	14.09	14.16

Table 4.2: Distances between each of the four natural initialization and their respective optimal solution for the four experiments

for an impressive result. The window and the algorithm experiment do not converge when running on a branch different from the first one—also the one that has the smallest distance—and when is running on the first converges in a few iterations. Hence, being able to provide a good initial transformation matrix W_0 and to correctly discriminate what are the best branches is vital for this approach.

These initial experiments show two different scenarios where these approaches are useful: either when the initial W_0 is good enough, or when there is some sort of supervision. The following experiments address these two cases.

4.2 The Iterative Hungarian as a Refinement Tool

Since a good initial transformation matrix W_0 is required in order to produce good results, it can be inferred that this iterative procedure could be used as a refinement tool. The experiments in this section perform the Iterative Hungarian starting with the initial condition W_0 produced from the following methods:

- The adversarial approach from Lample et al. (2017). This comprises the adversarial training described in Chapter 1 and a refinement step which consists of creating a dictionary from the best matches and then doing the supervised Procrustes.
- The supervised Procrustes approach.
- The Iterative Closest Point (ICP) method from Hoshen & Wolf (2018).

The experimental setup is exactly the same as in Lample et al. (2018): the code used for the adversarial part, the refinements, the dataset and the evaluation.[3] From that implementation we have also used the Procrustes method. The code for the ICP can be found in https://github.com/facebookresearch/Non-adversarialTranslation. In the three cases, the methodology has been the same: W_0 was obtained from each method, then we ran IH. We observed that the refinement step from Lample et al. (2017) improved results and we decided to use it as part of the method. CSLS with k=10 has been used for doing all the mappings and the accuracy reported corresponds to MUSE's evaluation (Lample et al. 2017).

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

	en-es	es-en	en-fr	fr-en	en-it	it-en	en-de	de-en	en-ru	ru-en	mean
MUSE (1)	**82.6**	83.7	82.5	82	76.8	77.6	**75.1**	72.5	**42.5**	60.1	73.54
MUSE (1) + us	**82.5**	**84.1**	**82.7**	**82.4**	**78.3**	**77.9**	**74.9**	**73.3**	**44.5**	**60.7**	**74.13**
MUSE (2)	81.9	83.2	82.1	82.4	**77.5**	77.5	74.7	72.9	**37**	61.9	73.11
MUSE (2) + us	**82.5**	**84.1**	**82.7**	**82.4**	**77.3**	**78.1**	**74.7**	**73.3**	**42.3**	**62.5**	**73.99**
MUSE (3)	82.1	**84**	82.1	82.3	**77.9**	77.7	74.8	69.9	37.1	60.1	72.8
MUSE (3) + us	**82.3**	**83.9**	**82.6**	**82.4**	**77.8**	**77.8**	**75.1**	**72.9**	**38.9**	**62.1**	**73.58**

Table 4.3: Accuracy for the unsupervised translation of different languages. IH run having as initialization matrix the transformation matrix W from MUSE over different seeds, and was then refined.

	en-es	es-en	en-fr	fr-en	en-it	it-en	en-de	de-en	en-ru	ru-en	mean
Procrustes	81.7	83.3	82.1	81.9	77.3	77.0	73.7	72.7	**49.9**	60.8	74.04
Procrustes + us	**82.5**	**84.2**	**82.2**	**82.6**	**78.1**	**78.0**	**75.0**	**73.5**	47.9	**63.9**	**74.79**

Table 4.4: Accuracy for the unsupervised translation of different languages. IH run having as initialization matrix the transformation matrix W from the supervised Procrustes, and was then refined.

	en-es	es-en	en-fr	fr-en	en-it	it-en	en-de	de-en	en-ru	ru-en	mean
ICP (1)	81.9	82.7	81.9	81.5	76.0	75.5	72.3	72.3	**46.4**	56.6	72.71
ICP (1) + us	**82.5**	**84.1**	**82.1**	**82.7**	**78.1**	**78.0**	**76.6**	**72.7**	46.2	**63.2**	**74.62**
ICP (2)	80.8	82.5	81.3	80.4	76.3	76.3	72.3	72.4	46.5	57.5	72.63
ICP (2) + us	**82.2**	**84.1**	**82.4**	**82.3**	**78.2**	**77.9**	**76.4**	**73.3**	**46.6**	**63.1**	**74.65**
ICP (3)	82.0	82.6	82.0	81.8	75.7	76.6	73.1	72.6	45.1	56.2	72.77
ICP (3) + us	**82.5**	**84.2**	82.0	**82.4**	**77.7**	**77.7**	**76.9**	**73.5**	**45.2**	**63.1**	**74.52**

Table 4.5: Accuracy for the unsupervised translation of different languages. IH run having as initialization matrix the transformation matrix W from ICP over different seeds, and was then refined.

The transformation matrix obtained from Lample et al. (2018) was trained on 200,000 words. Then we ran IH on a subsample of 45,000 words. Finally, the new transformation matrix was refined following the procedure in Lample et al. (2018). Also inspired by their work, the maps were calculated through Cross-Domain Similarity Local Scaling (CSLS) with 10 nearest neighbors.

We ran IH after a normalization of the word embeddings, which was found to achieve faster solutions. It must be noted that, since the adversarial part does not normalize word embeddings, the W_0 does not exactly correspond and not normalizing should provide better results at a higher computational cost. Hartmann et al. (2019) showed that unit length normalization makes GAN-based methods to become more unstable and deteriorates its performance, but supervised alignments or Procrustes refinement are not affected by it.

Results can be seen in Table 4.3 (MUSE), Table 4.4 (Procrustes) and Table 4.5 (ICP). It can be seen that indeed, this method improves the accuracy when used as a refinement tool. This is coherent with the fact that the other methods do not directly try to optimise the Wasserstein-Procrustes objective, although they achieve very good translations without relying on it.

Algorithm 3 Iterative Hungarian with a dictionary

Given a dictionary X_0, Y_0
$W_0 \leftarrow$ Procrustes (X_0, Y_0)
$X \leftarrow W_0$
$\left\|X_{NEW}^T Y_{NEW}\right\|_* = \infty$
while $\left\|X^T P Y\right\|_* < \left\|X_{NEW}^T Y_{NEW}\right\|_*$ **do**
  $\left\|X^T P Y\right\|_* \leftarrow \left\|X_{NEW}^T Y_{NEW}\right\|_*$
  $X \leftarrow X_{NEW}$
  $Y \leftarrow Y_{NEW}$
  for x, y **in** EquivalentProblems (X, Y) **do**
    $P \leftarrow$ Hungarian (x, y)
    if $\left\|x^T P y\right\|_*^2 < \left\|X_{NEW}^T Y_{NEW}\right\|_*$ **then**
      $W \leftarrow$ Procrustes (x, Py)
      $X_{NEW} \leftarrow xW$
      $Y_{NEW} \leftarrow Py$
      $\left\|X_{NEW}^T Y_{NEW}\right\|_* \leftarrow \left\|x^T P y\right\|_*^2$
    end if
  end for
end while

4.3 Adding supervision for language translation

Judging by the results in Table 4.1 we suggest that IH could be used for supervised translation. We repeated the same setup as in Section 4.1 but increasing the number of words ($n = 20,000$ and $n = 30,000$). In both cases it achieved a correct accuracy (100%) with 10% supervision.

We used a variant of IH for translation of English to Spanish using the same datasets and validation scores as in Lample et al. (2018). Our algorithm presents here a variant: in the first iteration W_0 is taken from the known dictionary as in the Procrustes problem. Then, for each iteration, the map between the words in the dictionary is not modified by the Hungarian algorithm. Finally, the new transformation matrix W_{n+1} is calculated solving the Procrustes problem on all the words (see Algorithm 3). However, there is another basic difference with respect to the experiments in Section 4.1: the solution of these problems is a permutation matrix (as they were English data trained in different ways), whereas the most accurate map from English to Spanish has not a one-to-one correspondence.

We ran tests of this setup on 10,000 words with different sizes for dictionaries. We qualitatively observed that, when the size of the dictionary is big (more than 10% of the data) and it induces a good initialization W_0 then the algorithm was not able to improve these results by a wide margin. When the size of the dictionary is small and it induces a bad accuracy score for W_0 (30%) then the final performance of the algorithm can either maintain that accuracy or go down. Finally, there is an intermediate region where this setup can improve the final accuracy. In particular, we observed that with that particular setup and a dictionary of only 5% of the words it was possible to achieve an accuracy of 68%, which could be refined up to 81% after doing the same refinements as in Lample et al. (2018).

Chapter 5

Related Work

Although there has been extensive work on bilingual lexicon induction (Hartmann et al. 2019), there have not been many approaches framing the problem as an optimal transport problem. Haghighi et al. (2008) proposed a self-learning method for bilingual lexicon induction, representing words with orthographic and context features and using the Hungarian algorithm to find an optimal 1:1 matching.

With the appearance of word embeddings, words were interpreted as vectors in high-dimensional spaces and concepts such as distance between words started to gain attention. Ruder et al. (2018) presented Viterbi EM: an approach where words were mapped following a one-to-one map and the isometries were induced by an orthogonal matrix. They deviated from the Wasserstein-Procrustes objective including a penalization term for unmatched words. Furthermore, they did not consider all possible matching -restricting to the k nearest neighbors- when running the Jonker Volgenant algorithm for the optimal transport problem.

Zhang et al. (2017) proposed two different methods: WGAN (an adversarial network that optimised Wasserstein distance) and EMDOT -an iterative procedure that used both Procrustes and solving a linear transport problem-. However, they considered the Earth Mover's Distance (EMD), which defines a distance between probability distributions, and applied to frequencies of words. They found that, although EMDOT could converge to bad local minima, it improved results when placed after WGAN, being the precursor for a refinement tool.

One-to-one maps have been considered (Ruder et al. 2018) and the orthogonal matrix has been used in many works, despite Søgaard et al. (2018) demonstrated that monolingual embedding spaces are not approximately isomorphic and that there is a complex relationship between word form and meaning. Grave et al. (2019) used Wasserstein-Procrustes as the objective function. They suggest an iterative procedure whose initial condition minimizes the convex Gold-Rangarajan relaxation with the Frank-Wolfe algorithm. The solution to this relaxation is then used as the initial condition for a gradient-based iterative procedure that stochastically samples different subsets of words for which there is not necessarily a direct translation, deviating from the Wasserstein-Procrustes objective.

Alaux et al. (2019) exploit the concept of Wasserstein-Procrustes objective for aligning multiple languages to a common space. They use, however, a different approach from ours: they minimise a loss function based on the CSLS matric from Lample et al. (2018). In a similar fashion, the entropic regularization of the Gromov-Wasserstein problem (Mémoli 2011) has been used for bilingual lexicon induction. Alvarez-Melis & Jaakkola (2018) focus on solving the problem through a Gromov-Wasserstein perspective.

Chapter 6

Conclusion and Future Work

We have presented work in rethinking the Wasserstein-Procrustes problem formulation for the task of aligning word embeddings across languages. In particular, we have demonstrated how properties of problems equivalent to Wasserstein-Procrustes can help in the unsupervised setup. We further showed that, in the semi-supervised setup, using just a little supervision can yield good results, especially if the datasets are similar or in the same language. We believe that our rethinking of the Wasserstein-Procrustes problem would enable further research and would eventually help develop better algorithms for aligning word embeddings across languages, especially if it is taken into account that most of the unsupervised approaches try to minimise loss functions different than Objective 1.2.

Some modifications of our approach induce interesting ideas for future work: In the same way Equation 2.11 is only dependent on the permutation matrix P, it is possible to develop a similar expression that involves W. Using the exact same properties, we obtain the following:

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle P, XWY^T\rangle_F = \tag{6.1}$$

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle P, U\Sigma V^T\rangle_F = \tag{6.2}$$

$$\underset{P\in\pi(N),W\in O(d)}{\arg\max} \langle S, \Sigma\rangle_F = \tag{6.3}$$

$$\underset{W\in O(d)}{\arg\max} \left\|XWY^T\right\|_* \tag{6.4}$$

but in this case, we assume that $U^TPV = I$, which is generally false. Nevertheless, we can use the orthogonal matrix $\widetilde{P} = UV^T$ and project it to the space of permutation matrices. This formulation of the problem has the advantage that the orthogonal matrix W generally has a lower dimensionality than the permutation matrix P, and thus it could be reduced through principal component analysis. We plan to explore this direction in future work.

A completely different approach would consider Objective 1.2 in the alternative form (2.7) and do an iterative procedure on the trace norm. From the property of the trace norm

$$\|A\|_* = \sup_{\|B\|\leq 1} |\operatorname{Tr}(BA)| \tag{6.5}$$

Hence, it is possible to design an iterative algorithm that optimizes B and $A = X^TPY$ jointly. Given a particular B_n, $\operatorname{Tr}\big(B_nX^TPY\big)$ is maximised through the Hungarian algorithm. Given one particular P_n, B can be calculated in a similar way and then be refined using Sinkhorn's theorem: given one matrix A with strictly positive elements, then there exist diagonal matrices D_1 and D_2 with strictly positive diagonal elements such that D_1AD_2 is doubly stochastic. The matrices D_1 and D_2 are unique modulo multiplying the first matrix by a positive number and dividing the second one by the same number (Sinkhorn 1964). A very simple application of this theorem is that, given one matrix A with stricly positive elements, the algorithm that alternatively rescales all rows and all columns of A to sum to 1 converges into a doubly stochastic matrix (Sinkhorn & Knopp 1967).

In conclusion, some methods achieve good results in the unsupervised translation of word embeddings without directly considering Problem 1.2 in the loss function (Lample et al. 2018, Grave et al. 2019, Alvarez-Melis & Jaakkola 2018, Hoshen & Wolf 2018). We have showed that there are a lot of approaches tackling Problem 1.2 and its interesting alternative formulations.

We have developed some mathematical properties of the Problem 1.2 and have used the concept of the different natural initialization transformations in an iterative algorithm. We have showed that this method achieves good results in the mapping of word embeddings from similar corpora. This method also has applications in word translation across different languages. In particular, we showed that it is possible to use this algorithm as a refinement tool and showed an improvement of results after using as the initialization matrix W_0 the transformation obtained by Lample et al. (2018). Finally, we showed how in our approach we can benefit from having a dictionary in a supervised approach.

Bibliography

Alaux, J., Grave, E., Cuturi, M. & Joulin, A. (2019), Unsupervised hyper-alignment for multilingual word embeddings, *in* 'International Conference on Learning Representations'.
URL: *https://openreview.net/forum?id=HJe62s09tX*

Alvarez-Melis, D. & Jaakkola, T. (2018), Gromov-Wasserstein alignment of word embedding spaces, *in* 'Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Brussels, Belgium, pp. 1881–1890.
URL: *https://www.aclweb.org/anthology/D18-1214*

Artetxe, M., Labaka, G. & Agirre, E. (2016), Learning principled bilingual mappings of word embeddings while preserving monolingual invariance, *in* 'Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Austin, Texas, pp. 2289–2294.
URL: *https://www.aclweb.org/anthology/D16-1250*

Artetxe, M., Labaka, G. & Agirre, E. (2017), Learning bilingual word embeddings with (almost) no bilingual data, *in* 'Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', Association for Computational Linguistics, Vancouver, Canada, pp. 451–462.
URL: *https://www.aclweb.org/anthology/P17-1042*

Artetxe, M., Labaka, G. & Agirre, E. (2018), A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings, *in* 'Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', Association for Computational Linguistics, Melbourne, Australia, pp. 789–798.
URL: *https://www.aclweb.org/anthology/P18-1073*

Barone, M. & Valerio, A. (2016), Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders, *in* 'Proceedings of the 1st Workshop on Representation Learning for NLP', Association for Computational Linguistics, Berlin, Germany, pp. 121–126.
URL: *https://www.aclweb.org/anthology/W16-1614*

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2017), 'Enriching word vectors with subword information', *Transactions of the Association for Computational Linguistics* **5**, 135–146.

Chen, K., Mikolov, T., Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space'.
URL: *http://arxiv.org/abs/1301.3781*

Cuturi, M. (2013), Sinkhorn distances: Lightspeed computation of optimal transport, *in* C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani & K. Q. Weinberger, eds, 'Advances in Neural Information Processing Systems 26', Curran Associates, Inc., pp. 2292–2300.
URL: *http://papers.nips.cc/paper/4927-sinkhorn-distances-lightspeed-computation-of-optimal-transport.pdf*

Dinu, G. & Baroni, M. (2014), 'Improving zero-shot learning by mitigating the hubness problem', *CoRR* **abs/1412.6568**.

Edmonds, J. & Karp, R. M. (1972), 'Theoretical improvements in algorithmic efficiency for network flow problems', *J. ACM* **19**(2), 248–264.
URL: *https://doi.org/10.1145/321694.321699*

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), Generative adversarial nets, *in* 'Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2', NIPS'14, MIT Press, Cambridge, MA, USA, p. 2672–2680.

Grave, E., Joulin, A. & Berthet, Q. (2019), Unsupervised alignment of embeddings with Wasserstein Procrustes, *in* 'The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan', pp. 1880–1890.
URL: *http://proceedings.mlr.press/v89/grave19a.html*

Haghighi, A., Liang, P., Berg-Kirkpatrick, T. & Klein, D. (2008), Learning bilingual lexicons from monolingual corpora, *in* 'Proceedings of ACL-08: HLT', Association for Computational Linguistics, Columbus, Ohio, pp. 771–779.
URL: *https://www.aclweb.org/anthology/P08-1088*

Hartmann, M., Kementchedjhieva, Y. & Søogaard, A. (2019), Comparing unsupervised word translation methods step by step, *in* 'Advances in Neural Information Processing Systems 32', Curran Associates, Inc., pp. 6033–6043.
URL: *http://papers.nips.cc/paper/8836-comparing-unsupervised-word-translation-methods-step-by-step.pdf*

Heinzerling, B. & Strube, M. (2018), BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages, *in* 'Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)', European Language Resources Association (ELRA), Miyazaki, Japan.
URL: *https://www.aclweb.org/anthology/L18-1473*

Hoshen, Y. & Wolf, L. (2018), Non-adversarial unsupervised word translation, *in* 'Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Brussels, Belgium, pp. 469–478.
URL: *https://www.aclweb.org/anthology/D18-1043*

Irvine, A. & Callison-Burch, C. (2013), Supervised bilingual lexicon induction with multiple monolingual signals, *in* 'Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013)'.

Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2017), Bag of tricks for efficient text classification, *in* 'Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers', Association for Computational Linguistics, pp. 427–431.

Lample, G., Conneau, A., Denoyer, L. & Ranzato, M. (2017), 'Unsupervised machine translation using monolingual corpora only', *arXiv preprint arXiv:1711.00043* .

Lample, G., Conneau, A., Ranzato, M., Denoyer, L. & Jégou, H. (2018), Word translation without parallel data, *in* 'International Conference on Learning Representations'.
URL: *https://openreview.net/forum?id=H196sainb*

Le, Q. V., Mikolov, T. & Sutskever, I. (2013), 'Exploiting similarities among languages for machine translation', *CoRR* **abs/1309.4168**.
URL: *http://arxiv.org/abs/1309.4168*

Mémoli, F. (2011), 'Gromov-Wasserstein distances and the metric approach to object matching', *Foundations of Computational Mathematics* **11**, 417–487.

Pennington, J., Socher, R. & Manning, C. D. (2014), Glove: Global vectors for word representation, *in* 'Empirical Methods in Natural Language Processing (EMNLP)', pp. 1532–1543.
URL: *http://www.aclweb.org/anthology/D14-1162*

Radovanovic, M., Nanopoulos, A. & Ivanovic;, M. (2010), 'Hubs in space: Popular nearest neighbors in high-dimensional data', *Journal of Machine Learning Research* **11**(86), 2487–2531.
URL: *http://jmlr.org/papers/v11/radovanovic10a.html*

Ruder, S., Cotterell, R., Kementchedjhieva, Y. & Søgaard, A. (2018), A discriminative latent-variable model for bilingual lexicon induction, *in* 'Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Brussels, Belgium, pp. 458–468.
URL: *https://www.aclweb.org/anthology/D18-1042*

Schönemann, P. H. (1966), 'A generalized solution of the orthogonal Procrustes problem', *Psychometrika* **31**(1), 1–10.
URL: *https://doi.org/10.1007/BF02289451*

Sinkhorn, R. (1964), 'A relationship between arbitrary positive matrices and doubly stochastic matrices', *Ann. Math. Statist.* **35**(2), 876–879.
URL: *https://doi.org/10.1214/aoms/1177703591*

Sinkhorn, R. & Knopp, P. (1967), 'Concerning nonnegative matrices and doubly stochastic matrices.', *Pacific J. Math.* **21**(2), 343–348.
URL: *https://projecteuclid.org:443/euclid.pjm/1102992505*

Smith, S. L., Turban, D. H. P., Hamblin, S. & Hammerla, N. Y. (2017), Offline bilingual word vectors, orthogonal transformations and the inverted softmax, *in* 'Proceedings of the 2017 International Conference on Learning Representations'.

Søgaard, A., Ruder, S. & Vulić, I. (2018), On the limitations of unsupervised bilingual dictionary induction, *in* 'Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', Association for Computational Linguistics, Melbourne, Australia, pp. 778–788.
URL: *https://www.aclweb.org/anthology/P18-1072*

Sutskever, I., Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), 'Distributed representations of words and phrases and their compositionality', *Advances in Neural Information Processing Systems* **26**.

Tomizawa, N. (1971), 'On some techniques useful for solution of transportation network problems', *Networks* **1**, 173–194.

Xing, C., Wang, D., Liu, C. & Lin, Y. (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', Association for Computational Linguistics, Denver, Colorado, pp. 1006–1011.
URL: *https://www.aclweb.org/anthology/N15-1104*

Zhang, M., Liu, Y., Luan, H. & Sun, M. (2017), Earth mover's distance minimization for unsupervised bilingual lexicon induction, *in* 'Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Copenhagen, Denmark, pp. 1934–1945.
URL: *https://www.aclweb.org/anthology/D17-1207*

Zhang, Y., Gaddy, D., Barzilay, R. & Jaakkola, T. (2016), Ten pairs to tag – multilingual POS tagging via coarse mapping between embeddings, *in* 'Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies', Association for Computational Linguistics, San Diego, California, pp. 1307–1317.
URL: *https://www.aclweb.org/anthology/N16-1156*