Loss in Translation:
Learning Bilingual Word Mapping with a Retrieval Criterion

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou and Edouard Grave

Facebook AI Research

`{ajoulin,bojanowski,tmikolov,rvj,egrave}@fb.com`

Abstract

Continuous word representations learned separately on distinct languages can be aligned so that their words become comparable in a common space. Existing works typically solve a quadratic problem to learn a orthogonal matrix aligning a bilingual lexicon, and use a retrieval criterion for inference. In this paper, we propose an unified formulation that directly optimizes a retrieval criterion in an end-to-end fashion. Our experiments on standard benchmarks show that our approach outperforms the state of the art on word translation, with the biggest improvements observed for distant language pairs such as English-Chinese.

1 Introduction

Previous work has proposed to learn a linear mapping between continuous representations of words by employing a small bilingual lexicon as supervision. The transformation generalizes well to words that are not observed during training, making it possible to extend the lexicon. Another application is to transfer predictive models between languages (Klementiev et al., 2012).

The first simple method proposed by Mikolov *et al.* (2013b) has been subsequently improved by changing the problem parametrization. One successful suggestion is to ℓ_2–normalize the word vectors and to constrain the linear mapping to be orthogonal (Xing et al., 2015). An alignment is then efficiently found using orthogonal Procrustes (Artetxe et al., 2016; Smith et al., 2017), improving the accuracy on standard benchmarks.

Yet, the resulting models suffer from the so-called "hubness problem": some word vectors tend to be the nearest neighbors of an abnormally high number of other words. This limitation is now addressed by applying a corrective metric at inference time, such as the inverted softmax (ISF) (Smith et al., 2017) or the cross-domain similarity local

scaling (CSLS) (Conneau et al., 2017). This is not fully satisfactory because the loss used for inference is not consistent with that employed for training. This observation suggests that the square loss is suboptimal and could advantageously be replaced by a loss adapted to retrieval.

In this paper, we propose a training objective inspired by the CSLS retrieval criterion. We introduce convex relaxations of the corresponding objective function, which are efficiently optimized with projected subgradient descent. This loss can advantageously include unsupervised information and therefore leverage the representations of words not occurring in the training lexicon.

Our contributions are as follows. First we introduce our approach and empirically evaluate it on standard benchmarks for word translation. We obtain state-of-the-art bilingual mappings for more than 25 language pairs. Second, we specifically show the benefit of our alternative loss function and of leveraging unsupervised information. Finally, we show that with our end-to-end formulation, a non-orthogonal mapping achieves better results. The code for our approach is a part of the fastText library[1] and the aligned vectors are available on `https://fasttext.cc/`.

2 Preliminaries on bilingual mappings

This section introduces pre-requisites and prior works to learn a mapping between two languages, using a small bilingual lexicon as supervision.

We start from two sets of continuous representations in two languages, each learned on monolingual data. Let us introduce some notation. Each word $i \in \{1, \ldots, N\}$ in the source language (respectively target language) is associated with a vector $\mathbf{x}_i \in \mathbb{R}^d$ (respectively $\mathbf{y}_i \in \mathbb{R}^d$). For simplicity,

---

[1] `https://github.com/facebookresearch/fastText/tree/master/alignment/`

we assume that our initial lexicon, or seeds, corresponds to the first n pairs $(\mathbf{x}_i, \mathbf{y}_i)_{i \in \{1,\dots,n\}}$. The goal is to extend the lexicon to all source words $i \in \{n+1,\dots,N\}$ that are not seeds. Mikolov *et al.* (2013b) learn a linear mapping $\mathbf{W} \in \mathbb{R}^{d \times d}$ between the word vectors of the seed lexicon that minimizes a measure of discrepancy between mapped word vectors of the source language and word vectors of the target language:

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times d}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{W}\mathbf{x}_i, \mathbf{y}_i), \qquad (1)$$

where ℓ is a loss function, typically the square loss $\ell_2(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$. This leads to a least squares problem, which is solved in closed form.

Orthogonality. The linear mapping $\mathbf{W}$ is constrained to be orthogonal, *i.e.* such that $\mathbf{W}^\top \mathbf{W} = \mathbf{I}_d$, where $\mathbf{I}_d$ is the d-dimensional identity matrix. This choice preserves distances between word vectors, and likewise word similarities. Previous works (Xing et al., 2015; Artetxe et al., 2016; Smith et al., 2017) experimentally observed that constraining the mapping in such a way improves the quality of the inferred lexicon. With the square loss and by enforcing an orthogonal mapping $\mathbf{W}$, Eq. (1) admits a closed form solution (Gower and Dijksterhuis, 2004): $\mathbf{W}^* = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}\mathbf{D}\mathbf{V}^\top$ is the singular value decomposition of the matrix $\mathbf{Y}^\top \mathbf{X}$.

Inference. Once a mapping $\mathbf{W}$ is learned, one can infer word correspondences for words that are not in the initial lexicon. The translation $t(i)$ of a source word i is obtained as

$$t(i) \in \operatorname*{arg\,min}_{j \in \{1,\dots,N\}} \ell(\mathbf{W}\mathbf{x}_i, \mathbf{y}_j). \qquad (2)$$

When the squared loss is used, this amounts to computing $\mathbf{W}\mathbf{x}_i$ and to performing a nearest neighbor search with respect to the Euclidean distance:

$$t(i) \in \operatorname*{arg\,min}_{j \in \{1,\dots,N\}} \|\mathbf{W}\mathbf{x}_i - \mathbf{y}_j\|_2^2. \qquad (3)$$

Hubness. A common observation is that nearest neighbor search for bilingual lexicon inference suffers from the "hubness problem" (Doddington et al., 1998; Dinu et al., 2014). Hubs are words that appear too frequently in the neighborhoods of other words. To mitigate this effect, a simple solution is to replace, at inference time, the square ℓ_2-norm in Eq. (3) by another criterion, such as ISF (Smith et al., 2017) or CSLS (Conneau et al., 2017).

This solution, both with ISF and CSLS criteria, is applied with a transformation $\mathbf{W}$ learned using the square loss. However, replacing the loss in Eq. (3) creates a discrepancy between the learning of the translation model and the inference.

3 Word translation as a retrieval task

In this section, we propose to directly include the CSLS criterion in the model in order to make learning and inference consistent. We also show how to incorporate unsupervised information..

The CSLS criterion is a similarity measure between the vectors $\mathbf{x}$ and $\mathbf{y}$ defined as:

$$\text{CSLS}(\mathbf{x}, \mathbf{y}) = -2\cos(\mathbf{x}, \mathbf{y})$$
$$+\frac{1}{k} \sum_{\mathbf{y}' \in \mathcal{N}_Y(\mathbf{x})} \cos(\mathbf{x}, \mathbf{y}') + \frac{1}{k} \sum_{\mathbf{x}' \in \mathcal{N}_X(\mathbf{y})} \cos(\mathbf{x}', \mathbf{y}),$$

where $\mathcal{N}_Y(\mathbf{x})$ is the set of k nearest neighbors of the point $\mathbf{x}$ in the set of target word vectors $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$, and $\cos$ is the cosine similarity. Note, the second term in the expression of the CSLS loss does not change the neighbors of $\mathbf{x}$. However, it gives a loss function that is symmetrical with respect to its two arguments, which is a desirable property for training.

Objective function. Let us now write the optimization problem for learning the bilingual mapping with CSLS. At this stage, we follow previous work and constrain the linear mapping $\mathbf{W}$ to belong to the set of orthogonal matrices $\mathcal{O}_d$. Here, we also assume that word vectors are ℓ_2-normalized. Under these assumptions, we have $\cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}_i) = \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_i$. Similarly, we have $\|\mathbf{y}_j - \mathbf{W}\mathbf{x}_i\|_2^2 = 2 - 2\mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_j$. Therefore, finding the k nearest neighbors of $\mathbf{W}\mathbf{x}_i$ among the elements of Y is equivalent to finding the k elements of Y which have the largest dot product with $\mathbf{W}\mathbf{x}_i$. We adopt this equivalent formulation because it leads to a convex formulation when relaxing the orthogonality constraint on $\mathbf{W}$. In summary, our optimization problem with the Relaxed CSLS loss (RCSLS) is written as:

$$\min_{\mathbf{W} \in \mathcal{O}_d} \frac{1}{n} \sum_{i=1}^{n} -2\mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_i$$
$$+\frac{1}{k} \sum_{\mathbf{y}_j \in \mathcal{N}_Y(\mathbf{W}\mathbf{x}_i)} \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_j$$
$$+\frac{1}{k} \sum_{\mathbf{W}\mathbf{x}_j \in \mathcal{N}_X(\mathbf{y}_i)} \mathbf{x}_j^\top \mathbf{W}^\top \mathbf{y}_i. \qquad (4)$$

Convex relaxation. Eq. (4) involves the minimization of a non-smooth cost function over the manifold of orthogonal matrices $\mathcal{O}_d$. As such, it can be solved using manifold optimization tools (Boumal et al., 2014). In this work, we consider as an alternative to the set $\mathcal{O}_d$, its convex hull $\mathcal{C}_d$, *i.e.*, the unit ball of the spectral norm. We refer to this projection as the "Spectral" model. We also consider the case where these constraints on the alignment matrix are simply removed.

Having a convex domain allows us to reason about the convexity of the cost function. We observe that the second and third terms in the CSLS loss can be rewritten as follows:

$$\sum_{\mathbf{y}_j \in \mathcal{N}_k(\mathbf{W}\mathbf{x}_i)} \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_j = \max_{S \in \mathcal{S}_k(n)} \sum_{j \in S} \mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_j,$$

where $\mathcal{S}_k(n)$ denotes the set of all subsets of $\{1, \ldots, n\}$ of size k. This term, seen as a function of $\mathbf{W}$, is a maximum of linear functions of $\mathbf{W}$, which is convex (Boyd and Vandenberghe, 2004). This shows that our objective function is convex with respect to the mapping $\mathbf{W}$ and piecewise linear (hence non-smooth). Note, our approach could be generalized to other loss functions by replacing the term $\mathbf{x}_i^\top \mathbf{W}^\top \mathbf{y}_j$ by any function convex in $\mathbf{W}$. We minimize this objective function over the convex set $\mathcal{C}_d$ by using the projected subgradient descent algorithm.

The projection onto the set $\mathcal{C}_d$ is solved by taking the singular value decomposition (SVD) of the matrix, and thresholding the singular values to one.

Extended Normalization. Usually, the number of word pairs in the seed lexicon n is small with respect to the size of the dictionaries N. To benefit from unlabeled data, it is common to add an iterative "refinement procedure" (Artetxe et al., 2017) when learning the translation model $\mathbf{W}$. Given a model $\mathbf{W}_t$, this procedure iterates over two steps. First it augments the training lexicon by keeping the best-inferred translation in Eq. (3). Second it learns a new mapping $\mathbf{W}_{t+1}$ by solving the problem in Eq. (1). This strategy is similar to standard semi-supervised approaches where the training set is augmented over time. In this work, we propose to use the unpaired words in the dictionaries as "negatives" in the RCSLS loss: instead of computing the k-nearest neighbors $\mathcal{N}_Y(\mathbf{W}\mathbf{x}_i)$ amongst the annotated words $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, we do it over the whole dictionary $\{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$.

4 Experiments

This section reports the main results obtained with our method. We provide complementary results and an ablation study in the appendix. We refer to our method without constraints as RCSLS and as RCSLS+spectral if the spectral constraints are used.

4.1 Implementation details

We choose a learning rate in $\{1, 10, 25, 50\}$ and a number of epochs in $\{10, 20\}$ on the validation set. For the unconstrained RCSLS, a small ℓ_2 regularization can be added to prevent the norm of $\mathbf{W}$ to diverge. In practice, we do not use any regularization. For the English-Chinese pairs (en-zh), we center the word vectors. The number of nearest neighbors in the CSLS loss is 10. We use the ℓ_2-normalized fastText word vectors by Bojanowski *et al.* (2017) trained on Wikipedia.

4.2 The MUSE benchmark

Table 1 reports the comparison of RCSLS with standard supervised and unsupervised approaches on 5 language pairs (in both directions) of the MUSE benchmark (Conneau et al., 2017). Every approach uses the Wikipedia fastText vectors and supervision comes in the form of a lexicon composed of 5k words and their translations. Regardless of the relaxation, RCSLS outperforms the state of the art by, on average, 3 to 4% in accuracy. This shows the importance of using the same criterion during training and inference. Note that the refinement step ("refine") also uses CSLS to finetune the alignments but leads to a marginal gain for supervised methods.

Interestingly, RCSLS achieves a better performance without constraints (+0.8%) for all pairs. Contrary to observations made in previous works, this result suggests that preserving the distance between word vectors is not essential for word translation. Indeed, previous works used a ℓ_2 loss where, indeed, orthogonal constraints lead to an improvement of +5.3% (Procrustes versus Least Square Error). This suggests that a linear mapping $\mathbf{W}$ with no constraints works well only if it is learned with a proper criterion.

Impact of extended normalization. Table 2 reports the gain brought by including words not in the lexicon (unannotated words) to the performance of RCSLS. Extending the dictionary significantly improves the performance on all language pairs.

Method	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en	avg.
Adversarial + refine	81.7	83.3	82.3	82.1	74.0	72.2	44.0	59.1	32.5	31.4	64.3
ICP + refine	82.2	83.8	82.5	82.5	74.8	73.1	46.3	61.6	-	-	-
Wass. Proc. + refine	82.8	84.1	82.6	82.9	75.4	73.3	43.7	59.1	-	-	-
Least Square Error	78.9	80.7	79.3	80.7	71.5	70.1	47.2	60.2	42.3	4.0	61.5
Procrustes	81.4	82.9	81.1	82.4	73.5	72.4	51.7	63.7	42.7	36.7	66.8
Procrustes + refine	82.4	83.9	82.3	83.2	75.3	73.2	50.1	63.5	40.3	35.5	66.9
RCSLS + spectral	83.5	85.7	82.3	**84.1**	78.2	75.8	56.1	66.5	44.9	45.7	70.2
RCSLS	**84.1**	**86.3**	**83.3**	**84.1**	**79.1**	**76.3**	**57.9**	**67.2**	**45.9**	**46.4**	**71.0**

Table 1: Comparison between RCSLS, Least Square Error, Procrustes and unsupervised approaches in the setting of Conneau et al. (2017). All the methods use the CSLS criterion for retrieval. "Refine" is the refinement step of Conneau et al. (2017). Adversarial, ICP and Wassertsein Proc. are unsupervised (Conneau et al., 2017; Hoshen and Wolf, 2018; Grave et al., 2018).

	en-es	en-fr	en-de	en-ru	avg.
Train	80.7	82.3	74.8	51.9	72.4
Ext.	**84.1**	**83.3**	**79.1**	**57.9**	**76.1**

Table 2: Accuracy with and without an extended normalization for RCSLS. "Ext." uses the full 200k vocabulary and "Train" only uses the pairs from the training lexicon.

	en-it	it-en
Adversarial + refine + CSLS	45.1	38.3
Mikolov et al. (2013b)	33.8	24.9
Dinu et al. (2014)	38.5	24.6
Artetxe et al. (2016)	39.7	33.8
Smith et al. (2017)	43.1	38.0
Procrustes + CSLS	44.9	**38.5**
RCSLS	**45.5**	38.0

Table 3: Accuracy on English and Italian with the setting of Dinu et al. (2014). "Adversarial" is an unsupervised technique. The adversarial and Procrustes results are from Conneau et al. (2017). We use a CSLS criterion for retrieval.

4.3 The WaCky dataset

Dinu et al. (2014) introduce a setting where word vectors are learned on the WaCky datasets (Baroni et al., 2009) and aligned with a noisy bilingual lexicon. We select the number of epochs within $\{1, 2, 5, 10\}$ on a validation set. Table 3 shows that RCSLS is on par with the state of the art. RCSLS is thus robust to relatively poor word vectors and noisy lexicons.

	Original			Aligned		
	Sem.	Synt.	Tot.	Sem.	Synt.	Tot.
Cs	26.4	76.7	63.7	**27.3**	**77.7**	**64.6**
De	**62.2**	56.9	**59.5**	61.4	**57.1**	59.3
Es	54.5	59.4	56.8	**55.1**	**61.1**	**57.9**
Fr	**76.0**	54.7	**68.5**	75.2	**55.1**	68.1
It	51.8	62.0	56.9	**52.7**	**63.8**	**58.2**
Pl	49.7	62.4	53.4	**50.9**	**63.2**	**54.5**
Zh	42.6	-	42.6	**47.2**	-	**47.2**
Avg.	51.9	62.0	57.3	**52.8**	**58.5**	**58.5**

Table 4: Performance on word analogies for different languages. We compare the original embeddings to their mapping to English. The mappings are learned using the full MUSE bilingual lexicons. We use the fastText vectors of Bojanowski et al. (2017).

4.4 Comparison with existing aligned vectors

Recently, word vectors based on fastText have been aligned and released by Smith *et al.* (2017, BabylonPartners, BP) and Conneau *et al.* (2017, MUSE). Both use a variation of Procrustes to align word vectors in the same space.

We compare these methods to RCSLS and report results in Table 5. RCSLS improves the performance by $+3.5\%$ over MUSE vectors when trained with the same lexicon (Original). Training RSCSL on the full training lexicon (Full) brings an additional improvement of $+2.9\%$ on average with a CSLS criterion, and $+6.1\%$ with a NN criterion. For reference, the performance of Procrustes only improves by $+1.4\%$ with CSLS and even degrades with a NN criterion. RCSLS benefits more from additional supervision than Procrustes. Finally,

	BP*	MUSE	Proc.	RCSLS	
		Orig.	Full	Orig.	Full
Bg	55.7	57.5	58.1	63.9	**65.2**
Ca	66.5	70.9	70.5	73.8	**75.0**
Cs	63.9	64.5	66.3	68.2	**71.1**
Da	66.8	67.4	68.3	71.1	**72.9**
De	68.9	72.7	73.5	76.9	**77.6**
El	54.9	58.5	60.1	62.7	**64.5**
Es	82.1	83.5	84.5	86.4	**87.1**
Et	41.5	45.7	47.3	49.5	**53.7**
Fi	56.7	59.5	61.9	65.8	**69.9**
Fr	81.7	82.4	82.5	**84.7**	84.7
He	51.5	54.1	55.4	57.8	**60.0**
Hr	48.9	52.2	53.4	55.6	**60.2**
Hu	61.9	64.9	66.1	69.3	**73.1**
Id	62.8	67.9	67.9	69.7	**72.9**
It	75.3	77.9	78.5	81.5	**82.8**
Mk	53.9	54.6	55.4	59.9	**60.4**
Nl	72.0	75.3	76.1	79.7	**80.5**
No	65.3	67.4	68.3	71.2	**73.3**
Pl	63.3	66.9	68.1	70.5	**73.5**
Pt	77.7	80.3	80.4	82.9	**84.6**
Ro	66.3	68.1	67.6	**74.0**	73.9
Ru	61.3	63.7	64.3	67.1	**70.3**
Sk	55.1	55.3	57.9	59.0	**61.7**
Sl	51.1	50.4	52.5	54.2	**58.2**
Sv	55.9	60.0	64.0	63.7	**69.5**
Tr	57.4	59.2	61.4	61.9	**65.8**
Uk	48.7	49.3	51.3	51.5	**55.5**
Vi	35.0	55.8	63.0	55.8	**66.9**
CSLS	60.8	63.8	65.2	67.4	**70.2**
NN	54.6	57.4	57.5	62.4	**68.5**

Table 5: Comparison with publicly available aligned vectors over 28 languages. All use supervision. Alignements are learned either on the "Original" or "Full" MUSE training. We report the detailed performance with a CSLS criterion and the average for both NN and CSLS criteria.
*BP uses a different training set of comparable size.

the gap between RCSLS and the other methods is higher with a NN criterion, suggesting that RCSLS imports some of the properties of CSLS to the dot product between aligned vectors.

4.5 Impact on word vectors

Non-orthogonal mapping of word vectors changes their dot products. We evaluate the impact of this mapping on word analogy tasks (Mikolov et al.,

		Original	Aligned
De	Gur350	72	**74**
	Ws350	68	**70**
	Zg222	**46**	44
Es	Ws353	59	**61**
It	Ws350	**64**	**64**
Pt	Ws353	**60**	58
Ro	Ws353	**58**	55
	Hj	**67**	65
	Ws350	**59**	58
Zh	Sim	35	**44**
Avg.		58.8	**59.2**

Table 6: Performance on word similarities for different languages. We compare the original embeddings to their mapping to English. The mappings are learned with the full MUSE bilingual lexicons over the fastText vectors of Bojanowski et al. (2017).

2013a). In Table 4, we report the accuracy on analogies for raw word vectors and our vectors mapped to English with an alignement trained on the full MUSE training set. Regardless of the source language, the mapping does not negatively impact the word vectors. Similarly, our alignement has also little impact on word similarity, as shown in Table 6.

We confirm this observation by running the reverse mapping, i.e., by mapping the English word vectors of Mikolov et al. (2018) to Spanish. It leads to an improvement of 1% both for vectors trained on Common Crawl (85% to 86%) and Wikipedia + News (87% to 88%).

5 Conclusion

This paper shows that minimizing a convex relaxation of the CSLS loss significantly improves the quality of bilingual word vector alignment. We use a reformulation of CSLS that generalizes to convex functions beyond dot products and provides a single end-to-end training that is consistent with the inference stage. Finally, we show that removing the orthogonality constraint does not degrade the quality of the aligned vectors.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 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*, pages 2289–2294.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 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)*, volume 1, pages 451–462.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

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. https://fasttext.cc/docs/en/pretrained-vectors.html.

Nicolas Boumal, Bamdev Mishra, P.-A. Absil, and Rodolphe Sepulchre. 2014. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

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

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*.

George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, and Douglas Reynolds. 1998. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. Technical report.

John C Gower and Garmt B Dijksterhuis. 2004. *Procrustes problems*, volume 30. Oxford University Press on Demand.

Edouard Grave, Armand Joulin, and Quentin Berthet. 2018. Unsupervised alignment of embeddings with wasserstein procrustes. *arXiv preprint arXiv:1805.11222*.

Yedid Hoshen and Lior Wolf. 2018. An iterative closest point method for unsupervised word translation. *arXiv preprint arXiv:1801.06126*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. *Proceedings of COLING 2012*, pages 1459–1474.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

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

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*. https://github.com/Babylonpartners/fastText_multilingual.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 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*, pages 1006–1011.