Characterizing Departures from Linearity in Word Translation

Ndapa Nakashole and **Raphael Flauger**
Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093
`nnakashole@eng.ucsd.edu`

Abstract

We investigate the behavior of maps learned by machine translation methods. The maps translate words by projecting between word embedding spaces of different languages. We locally approximate these maps using linear maps, and find that they vary across the word embedding space. This demonstrates that the underlying maps are non-linear. Importantly, we show that the locally linear maps vary by an amount that is tightly correlated with the distance between the neighborhoods on which they are trained. Our results can be used to test non-linear methods, and to drive the design of more accurate maps for word translation.

Figure 1: **Top:** Assumption of linearity implies a single linear map $\mathbf{M}$. **Bottom:** Our hypothesis is that the underlying map is expected to be non-linear but in small enough neighborhoods can be approximated by linear maps $\mathbf{M}^{x_i}$ for each neighborhood defined by $\mathbf{x_i}$.

1 Introduction

Following the success of monolingual word embeddings (Collobert et al., 2011), a number of studies have recently explored multilingual word embeddings. The goal is to learn word vectors such that similar words have similar vector representations regardless of their language (Zou et al., 2013; Upadhyay et al., 2016). Multilingual word embeddings have applications in machine translation, and hold promise for cross-lingual model transfer in NLP tasks such as parsing or part-of-speech tagging.

A class of methods has emerged whose core technique is to learn *linear* maps between vector spaces of different languages (Mikolov et al., 2013a; Faruqui and Dyer, 2014; Vulic and Korhonen, 2016; Artetxe et al., 2016; Conneau et al., 2018). These methods work as follows: For a given pair of languages, first, monolingual word vectors are learned independen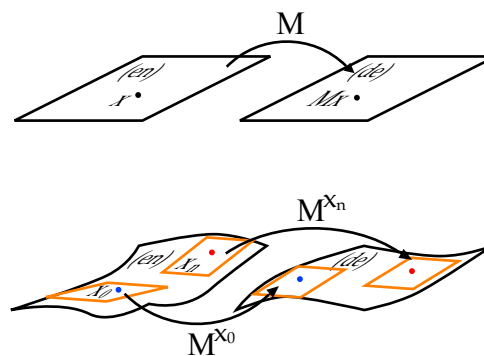tly for each language, and second, under the assumption that word vector spaces exhibit comparable structure across languages, a linear mapping function is learned to connect the two monolingual vector spaces. The map can then be used to translate words between the language pair.

Both seminal (Mikolov et al., 2013a), and state-of-the-art methods (Conneau et al., 2018) found linear maps to substantially outperform specific non-linear maps generated by feedforward neural networks. Advantages of linear maps include: *1)* In settings with limited training data, accurate linear maps can still be learned (Conneau et al., 2018; Zhang et al., 2017; Artetxe et al., 2017; Smith et al., 2017). For example, in unsupervised learning, (Conneau et al., 2018) found that using non-linear mapping functions made adversarial training unstable[1]. *2)* One can easily impose constraints on the linear maps at training time to ensure that the quality of the monolingual em-

---

[1] https://openreview.net/forum?id=H196sainb

beddings is preserved after mapping (Xing et al., 2015; Smith et al., 2017).

However, it is not well understood to what extent the assumption of linearity holds and how it affects performance. In this paper, we investigate the behavior of word translation maps, and show that there is clear evidence of departure from linearity.

Non-linear maps beyond those generated by feedforward neural networks have also been explored for this task (Lu et al., 2015; Shi et al., 2015; Wijaya et al., 2017; Shi et al., 2015). However, no attempt was made to characterize the resulting maps.

In this paper, we allow for an underlying mapping function that is non-linear, but assume that it can be approximated by linear maps at least in small enough neighborhoods. If the underlying map is linear, all local approximations should be identical, or, given the finite size of the training data, similar. In contrast, if the underlying map is non-linear, the locally linear approximations will depend on the neighborhood. Figure 1 illustrates the difference between the assumption of a single linear map, and our working hypothesis of locally linear approximations to a non-linear map. The variation of the linear approximations provides a characterization of the nonlinear map. We show that the local linear approximations vary across neighborhoods in the embedding space by an amount that is tightly correlated with the distance between the neighborhoods on which they are trained. The functional form of this variation can be used to test non-linear methods.

2 Review of Prior Work

To learn linear word translation maps, different loss functions have been proposed. The simplest is the regularized least squares loss, where the linear map $\mathbf{M}$ is learned as follows: $\hat{\mathbf{M}} = \arg\min_{\mathbf{M}} ||\mathbf{M}\mathbf{X} - \mathbf{Y}||_F + \lambda||\mathbf{M}||$, here $\mathbf{X}$ and $\mathbf{Y}$ are matrices that contain word embedding vectors for the source and target language (Mikolov et al., 2013a; Dinu et al., 2014; Vulic and Korhonen, 2016). The translation t of a source language word s is then given by: $t = \arg\max_t \cos(\mathbf{M}x_s, y_t)$.

(Xing et al., 2015) obtained improved results by imposing an orthogonality constraint on $\mathbf{M}$, $\mathbf{M} = ||\mathbf{M}\mathbf{W}^T - \mathbf{I}||$ where $\mathbf{I}$ is the identify matrix. Another loss function used in prior work
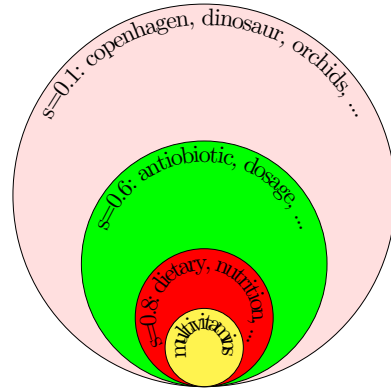

Figure 2: Neighborhoods formed around the word "multivitamins".

is the max-margin loss, which has been shown to significantly outperform the least squares loss (Lazaridou et al., 2015; Nakashole and Flauger, 2017).

Unsupervised or limited supervision methods for learning word translation maps have recently been proposed (Conneau et al., 2018; Zhang et al., 2017; Artetxe et al., 2017; Smith et al., 2017). However, the underlying methods for learning the mapping function are similar to prior work (Xing et al., 2015).

Non-linear cross-lingual mapping methods have been proposed. In (Wijaya et al., 2017) when dealing with rare words, the proposed method backs-off to a feed-forward neural network. (Shi et al., 2015) model relations across languages. (Lu et al., 2015) proposed a deep canonical correlation analysis based mapping method. Work on phrase translation has explored the use of many local maps that are individually trained (Zhao et al., 2015). In contrast to our work, these prior papers do not attempt to characterize the behavior of the resulting maps.

Our hypothesis is similar in spirit to the use of locally linear embeddings for nonlinear dimensionality reduction (Roweis and Saul, 2000).

3 Neighborhoods in Word Vector Space

In order to study the behavior of word translation maps, we begin by introducing a simple notion of neighborhoods in the embedding space. For a given language (e.g., English, en), we define a *neighborhood* of a word as follows: First, we pick a word x_i, whose corresponding vector is $x_i \in \mathbf{X}^{en}$, as an anchor. Second, we initialize a neighborhood $\mathcal{N}(x_i)$ containing a single

vector x_i. We then grow the neighborhood by adding all words whose cosine similarity to x_i is $\geq s$. The resulting neighborhood is defined as: $\mathcal{N}(x_i, s) = \{ x_j \mid \cos(x_i, x_j) \geq s \}$.

Suppose we pick the word *multivitamins* as the anchor word. We can generate neighborhoods using $\mathcal{N}(x_{\,multivitamins}, s)$ where for each value of s we get a different neighborhood. Neighborhoods corresponding to larger values of s are subsumed by those corresponding to smaller values of s.

Figure 2 illustrates the process of generating neighborhoods around the word *multivitamins*. For large values of s (e.g., $s = 0.8$), the resulting neighborhoods only contain words that are closely related to the anchor word, such as *dietary* and *nutrition*. As s gets smaller (e.g., $s = 0.6$), the neighborhood gets larger, and includes words that are less related to the anchor, such as *antibiotic*.

Using this simple method, we can define different-sized neighborhoods around any word in the vocabulary.

4 Analysis of Map Behavior

Given the above neighborhood definition, we now seek to understand how word translation maps change as we move across neighborhoods in word embedding space.

Questions Studied. We study the following questions: [Q.1] Is there a single linear map for word translation that produces the same level of performance regardless of where in the vector space the words being translated fall? [Q.2] If there is no such single linear map, but instead multiple neighborhood-specific ones, is there a relationship between neighborhood-specific maps and the distances between their respective neighborhoods?

4.1 Experimental Setup and Data

In our first experiment we translate from English (en) to German (de). We obtained pre-trained word embeddings from FastText (Bojanowski et al., 2017). In the first experiment, we follow common practice (Mikolov et al., 2013a; Ammar et al., 2016; Nakashole and Flauger, 2017; Vulic and Korhonen, 2016), and used the Google Translate API to obtain training and test data. We make our data available for reproducibility[2]. For the second experiment, we repeat the first experiment, but instead of using Google Translate, we

use the recently released Facebook AI Research dictionaries [3] for training and test data. To test the generality of the findings of the first experiment, the second experiment was performed on a different language pair: English (en) to Swedish (sv).

In our all experiments, the cross-lingual maps are learned using the max-margin loss, which has been shown to perform competitively, while having fast run-times. (Lazaridou et al., 2015; Nakashole and Flauger, 2017). The max-margin loss aims to rank correct training data pairs (x_i, y_i) higher than incorrect pairs (x_i, y_j) with a margin of at least γ. The margin γ is a hyper-parameter and the incorrect labels, y_j can be selected randomly such that $j \neq i$ or in a more application specific manner. In our experiments, we set $\gamma = 0.4$ and randomly selected negative examples, one negative example for each training data point.

Given a seed dictionary as training data of the form $D^{tr} = \{x_i, y_i\}_{i=1}^m$, the mapping function is

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_{i=1}^m \sum_{j \neq i}^k \max \Big(0, \gamma$$
$$+ d(y_i, \mathbf{W}x_i) - d(y_j, \mathbf{W}x_i) \Big), \qquad (1)$$

where $\hat{y}_i = \mathbf{W}x_i$ is the prediction, k is the number of incorrect examples per training instance, and $d(x, y) = (x - y)^2$ is the distance measure.

For the first experiment, we picked the following words as anchor words and obtained maps associated with each of their neighborhoods: $\mathbf{M}^{(multivitamins)}$, $\mathbf{M}^{(antibiotic)}$, $\mathbf{M}^{(disease)}$, $\mathbf{M}^{(blowflies)}$, $\mathbf{M}^{(dinosaur)}$, $\mathbf{M}^{(orchids)}$, $\mathbf{M}^{(copenhagen)}$. For each anchor word, we set $s = 0.5$, thus the neighborhoods are $\mathcal{N}(x_i, 0.5)$ where x_i is the vector of the anchor word. The training data for learning each neighborhood-specific linear map consists of vectors in $\mathcal{N}(x_i, 0.5)$ and their translations.

Table 1 shows details of the training and test data for each neighborhood. The words shown in Table 1 were picked as follows: first we picked the word *multivitamins*, then we picked the other words to have varying degrees of similarity to it. The cosine similarity of these words to the word 'multivitamins' are shown in column 3 of Table 1. It is also worth noting that there is nothing special about these words. In fact, the second experiment

0	1	2	3	4	5	6	7	8	9
Anchor Word	Data, $\mathcal{N}(x_i, s = 0.5)$		x_0 Similarity	Translation Accuracy				Matrix Property	
	Train	Test	$\cos(x_0, x_i)$	$\mathbf{M}$	$\mathbf{M^{x_0}}$	$\mathbf{M^{x_i}}$	Δ	$\cos(\mathbf{M^{x_0}}, \mathbf{M^{x_i}})$	$\|M\|$
x_0:multivitamins	3,415	500	1.0	58.3	**68.2**	**68.2**	0	1.0	33.07
x_1:antibiotic	3,507	500	0.60	61.1	67.3	**72.7**	5.4 ↑	0.59	33.29
x_2:disease	2,478	500	0.45	69.3	59.2	**73.4**	14.2 ↑	0.31	35.35
x_3:blowflies	2,434	500	0.33	71.4	28.4	**73.2**	44.8 ↑	0.20	33.36
x_4:dinosaur	990	500	0.24	63.2	14.7	**77.1**	62.4 ↑	0.14	36.50
x_5:orchids	2,981	500	0.19	73.7	19.3	**78.0**	58.7 ↑	0.20	30.68
x_6:copenhagen	2,083	500	0.11	38.5	31.2	**67.4**	36.2 ↑	0.15	31.42

Table 1: The behavior of word translation maps trained on different neighborhoods (en $\rightarrow$ de translation). Highlighted columns illustrate variations in maps. Accuracy refers to precision at 10.

was carried out on different set of words, and on a different language pair.

4.2 Map Similarity Analysis

If indeed there exists a map that is the same linear map everywhere, we expect the above neighborhood-specific maps to be similar. Our analysis makes use of the following definition of matrix similarity:

$$\cos(\mathbf{M_1}, \mathbf{M_2}) = \frac{tr(\mathbf{M_1}^T \mathbf{M_2})}{\sqrt{tr(\mathbf{M_1}^T \mathbf{M_1}) tr(\mathbf{M_2}^T \mathbf{M_2})}} \quad (2)$$

Here $tr(\mathbf{M})$ denotes the trace of the matrix $\mathbf{M}$. $tr(\mathbf{M_1}^T \mathbf{M_1})$ computes the Frobenius norm $\|\mathbf{M_1}\|^2$, and $tr(\mathbf{M_1}^T \mathbf{M_2})$ is the Frobenius inner product. That is, $\cos(\mathbf{M_1}, \mathbf{M_2})$ computes the cosine similarity between the vectorized versions of matrices $\mathbf{M_1}$ and $\mathbf{M_2}$.

4.3 Experimental Results

The main results of our analysis are shown in Table 1.

We now analyze the results of Table 1 in detail. The *0th* column contains the anchor word, x_i, around which the neighborhood is formed. The *1st*, and *2nd* columns contain the size of the training and test data from $\mathcal{N}(x_i, s = 0.5)$ where x_i is the word vector for the anchor word.

The *3rd* column contains the cosine similarity between x_0, *multivitamins*, and x_i. For example, x_1 (*antibiotic*) is the most similar to x_0 (0.6), and x_6, *copenhagen*, is the least similar to x_0 (0.11).

The *4th* column is the translation accuracy of the single global map M, training on data from all x_i neighborhoods. The *5th* column is the translation accuracy of the map M^{x_0}, trained on the train-
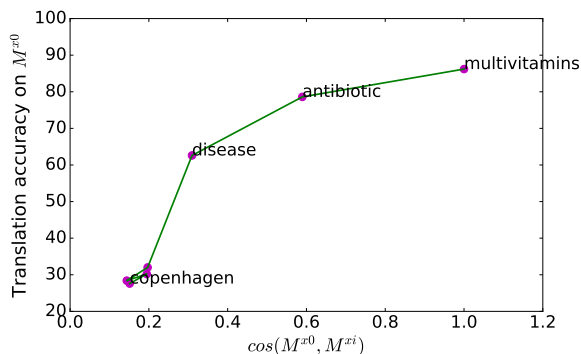

Figure 3: Correlation between the *8th* column (x-axis), map similarity $\cos(\mathbf{M^{x_0}}, \mathbf{M^{x_i}})$, and the *5th* column (y-axis), performance of map M^{x_0} on test data from the neighborhood anchored at x_i.

ing data of x_0, and tested on the test data in x_i. We use precision at top-10 as a measure of translation accuracy. Going down this column we can see that accuracy is highest on the test data from the neighborhood anchored at x_0 itself, and lowest on the test data from the neighborhood anchored at x_6, copenhagen, which is also the furthest word from x_0.

The *6th* column is translation accuracy of the map M^{x_i}, trained on the training data of the neighborhood anchored at x_i, and tested on the test data in x_i. We can see that compared to the *5th* column, in all cases performance is higher when we apply the map trained on data from the neighborhood, M^{x_i} instead of M^{x_0}. The *7th* column shows the difference in translation accuracy of the map M^{x_i} and M^{x_0}. This shows that the more dissimilar the neighborhood anchor word x_i is from x_0 according to the cosine similarity shown in the *4rd* column, the larger this difference is.

The local maps, *6th* column, M^{x_i} in all cases

0	1	2	3	4	5	6
Anchor Word	Data, $\mathcal{N}(x_i, s = 0.5)$		x_0 Similarity	Translation Accuracy		
	Train	Test	$\cos(x_0, x_i)$	$\mathbf{M}$	$\mathbf{M^{x_0}}$	$\mathbf{M^{x_i}}$
x_0:species	1,765	200	1.0	58.0	**60.0**	**60.0**
x_1:genus	1,374	200	0.77	56.4	53.1	**56.0**
x_2:laticeps	1,868	200	0.60	81.0	73.4	**79.1**
x_3:femoralis	1,689	200	0.52	85.3	76.3	**83.4**
x_4:coneflower	1,077	200	0.47	39.6	31.6	**43.0**
x_5:epicauta	1,339	200	0.43	53.4	42.1	**54.2**
x_6:kristoffersen	1,227	200	0.09	24.3	10.3	**57.1**

Table 2: Different language pair (en $\rightarrow$ sv) English to Swedish, and different sets of neighborhoods. Train and test is from the FAIR/MUSE word translation lexicons. Accuracy refers to precision at 10.

outperform the global map $4th$ column.

The *8th* column shows the similarity between maps M^{x_i} and M^{x_0} as computed by Equation 2. This column shows that the similarity between these learned maps is highly correlated with the cosine similarity or distance between the words in $3rd$ column. We also see a correlation with the translation accuracy in the $5th$ column. This correlation is visualized in Figure 3. Finally, the $9th$ column shows the magnitudes of the maps. The magnitudes vary somewhat between the maps trained on the different neighborhoods, and are significantly different from the magnitude expected for an orthogonal matrix. (For an orthogonal 300×300 matrix O the norm is $||O|| = \sqrt{300} \approx 17$).

In order to determine the generality of our results, we carried out the same experiment on a different language pair, as shown in Table 2. Crucially, we see the same trends as those observed in Table 1. This supports the generality of our findings.

4.4 Experiments Summary

Our experimental study suggests the following: i) linear maps vary across neighborhoods, implying that the assumption of a linear map does not to hold. ii) the difference between maps is tightly correlated with the distance between neighborhoods.

5 Conclusions

In this paper, we provide evidence that the assumption of linearity made by a large body of current work on cross-lingual mapping for word translation does not hold. We locally approximate the underlying non-linear map using linear maps, and show that these maps vary across neighborhoods in vector space by an amount that is tightly correlated with the distance between the neighborhoods on which they are trained. These results can be used to test non-linear methods, and we plan to use our finding to design more accurate maps in future work.

Acknowledgments

References

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *CoRR*, abs/1602.01925.

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.

Phil Blunsom and Karl Moritz Hermann. 2014. Multilingual distributed representations without word alignment.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*.

A. P. Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M. Khapra, Balaraman Ravindran, Vikas C. Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In *NIPS*, pages 1853–1861.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data.

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

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *EACL*, pages 462–471.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *IICML*, pages 748–756.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *NAACL*, pages 1386–1390.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *ACL*, pages 771–779.

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

Tomáš Kočiský, Karl Moritz Hermann, and Phil Blunsom. 2014. Learning bilingual word representations by marginalizing alignments. *arXiv preprint arXiv:1405.0947*.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *ACL Workshop on Unsupervised Lexical Acquisition*.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *ACL*, pages 270–280.

Jinyu Li, Rui Zhao, Jui-Ting Huang, and Yifan Gong. 2014. Learning small-size dnn with output-distribution-based criteria. In *INTERSPEECH*, pages 1910–1914.

Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Deep multilingual correlation for improved word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 250–256.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ndapandula Nakashole and Raphael Flauger. 2017. Knowledge distillation for bilingual dictionary induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2487–2496.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *ACL*.

Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

Tianze Shi, Zhiyuan Liu, Yang Liu, and Maosong Sun. 2015. Learning cross-lingual word embeddings via matrix co-factorization. In *Proceedings of the 53rd*

Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), volume 2, pages 567–572.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *ICLR*.

Anders Søgaard, Zeljko Agic, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual NLP. In *ACL*, pages 1713–1722.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Intell. Res. (JAIR)*, 37:141–188.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *ACL*.

Ivan Vulic and Anna Korhonen. 2016. On the role of seed lexicons in learning bilingual word embeddings. ACL.

Ivan Vulic and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *ACL*, pages 719–725.

Derry Tanti Wijaya, Brendan Callahan, John Hewitt, Jie Gao, Xiao Ling, Marianna Apidianaki, and Chris Callison-Burch. 2017. Learning translations via matrix completion. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1452–1463.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *HLT-NAACL*, pages 1006–1011.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1959–1970.

Kai Zhao, Hany Hassan, and Michael Auli. 2015. Learning translation models from monolingual continuous representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1527–1536.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398.