Why is unsupervised alignment of English embeddings from different algorithms so hard?

Mareike Hartmann
Dep. of Computer Science
University of Copenhagen
Denmark
hartmann@di.ku.dk

Yova Kementchedjhieva
Dep. of Computer Science
University of Copenhagen
Denmark
yova@di.ku.dk

Anders Søgaard
Dep. of Computer Science
University of Copenhagen
Denmark
soegaard@di.ku.dk

Abstract

This paper presents a challenge to the community: Generative adversarial networks (GANs) can perfectly align independent English word embeddings induced using *the same* algorithm, based on distributional information alone; but fails to do so, for two different embeddings algorithms. *Why is that?* We believe understanding why, is key to understand *both* modern word embedding algorithms *and* the limitations and instability dynamics of GANs. This paper shows that (a) in all these cases, where alignment fails, there exists a linear transform between the two embeddings (so algorithm biases do not lead to non-linear differences), and (b) similar effects can not easily be obtained by varying hyper-parameters. One plausible suggestion based on our initial experiments is that the differences in the inductive biases of the embedding algorithms lead to an optimization landscape that is riddled with local optima, leading to a very small basin of convergence, but we present this more as a challenge paper than a technical contribution.

1 Introduction

This paper brings together two fascinating research topics in natural language processing (NLP), namely *understanding the properties of word embeddings* (Mikolov et al., 2013; Mitchell and Steedman, 2015; Mimno and Thompson, 2017) and *unsupervised bilingual dictionary induction* (Conneau et al., 2018; Zhang et al., 2017; Søgaard et al., 2018). In an effort to better understand when unsupervised bilingual dictionary induction is possible, we factored out linguistic differences between languages, and studied English-English alignability (by learning to align English embeddings trained on different samples of the English Wikipedia), when we came across a puzzling phenomena: *English-English can be aligned with almost 100% precision, if you use the same embedding algorithms for the two samples, but not at all (0% precision), if you use different embedding algorithms*. This results suggest that the properties of word embeddings induced by different algorithms challenge unsupervised bilingual dictionary algorithms. Understanding why will enable us to develop more stable adversarial learning algorithms and give us a better understanding of how embedding algorithms differ.

Contributions We are, to the best of our knowledge, the first to study unsupervised alignability of pairs of English word embeddings. We show that unsupervised alignment – specifically the MUSE system (Conneau et al., 2018) – fails when the algorithms used to induce the two embeddings differ, and that this is *not* because there is no linear transformation between the two spaces. We further show that poor initialization, as a result of MUSE initially applying an identity transform to two word embeddings far apart in space, is not the sole reason the discriminator suffers from local optima. Finally, we present an experiment showing what the minimal corpus size is for unsupervised alignment to succeed, in the absence of linguistic differences.

2 Aligning embeddings

2.1 Unsupervised alignment using generative adversarial networks

MUSE (Conneau et al., 2018) uses a vanilla generative adversarial network (GAN) with a linear generator to learn alignments between embedding spaces without supervision. In a two-player game, a discriminator D aims to tell the two language spaces apart, while a generator G aims to map the source language into the target language space, fooling the discriminator. While MUSE achieves impressive results at times, MUSE is highly unstable, e.g., with different initializations precision

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 582–586
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

scores vary between 0% and 45% for English-Greek (Søgaard et al., 2018).

The parameters of a GAN with a linear generator are (Ω, w). They are obtained by solving the following min-max problem:

$$\min_{\Omega} \max_{w} \mathrm{E}[\log(D_w(X)) + \log(1 - D_w(g_\Omega(Z)))] \tag{1}$$

which reduces to

$$\min_{\Omega} JS(P_X \mid P_\Omega) \tag{2}$$

Ω is initialized as the identity matrix I.

If G wins the game against an ideal discriminator on a very large number of samples, then F (the source vector space) and ΩE (with E being the target vector space) can be shown to be close in Jensen-Shannon divergence, and thus the model has learned the true distribution. This result, referring to the distributions of the data, p_{data}, and the distribution, p_g, G is sampling from, is from Goodfellow et al. (2014): *If G and D have enough capacity, and at each step of training, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion*

$$E_{\mathbf{x}\sim p_{data}}[\log D^*_G(\mathbf{x})] + E_{\mathbf{x}\sim p_g}[\log(1 - D^*_G(\mathbf{x}))]$$

then p_g converges to p_{data}.

This result relies on a number of assumptions that do not hold in practice. Our generator, which learns a linear transform Ω, has very limited capacity, for example, and we are updating Ω rather than p_g. In practice, therefore, during training, we alternate between k steps of optimizing the discriminator and one step of optimizing the generator. If the GAN-based alignment is not successful, this can thus be a result of two things: Either that G does not have enough capacity, or that D is stuck in a local optimum. Our results in §3 show that the inability to align English-English in the case of different word embedding algorithms is *not* a result of limited capacity, but a result of the GAN being trapped in one of the many local optima of the loss function.

2.2 Supervised alignment using Procrustes Analysis

Procrustes Analysis (Schönemann, 1966) has been commonly used for supervised alignment of word embeddings (Smith et al., 2017; Artetxe et al., 2018). Here, the optimal alignment between two embedding spaces is computed using singular value decomposition of the aligned embeddings in a seed dictionary. Conneau et al. (2018) use Procrustes Analysis to refine an initial seed dictionary learned by the generative adversarial network without supervision. In our supervised experiments, we use 5000 seed words as supervision for learning the alignment between embeddings.

2.3 Geometry of embeddings

Below we summarize some previous findings about the geometry of monolingual embeddings (Mimno and Thompson, 2017), and add some new observations. We discuss five embedding algorithms: SVD on positive PMI matrices (Hyperwords-SVD) (Levy et al., 2015), skip-gram negative sampling applied to co-occurrence matrices (Hyperwords-SGNS) (Levy et al., 2015), continuous bag-of-words (CBOW) (Mikolov et al., 2013a), GloVe (Pennington et al., 2014), and FastText (Bojanowski et al., 2017). To analyze the geometry of our monolingual embeddings in space, we report average inner product to mean vector; see Mimno and Thompson (2017) for details.

Hyperwords-SVD have a small average inner product (0.0032), suggesting they are well-dispersed through space; like Hyperwords-SGNS and standard SGNS (Mimno and Thompson, 2017), they do not exhibit a clear word frequency bias. **Hyperwords-SGNS** vectors also have a small average inner product (0.0002), in contrast with standard SGNS vectors, which are narrowly clustered in a single orthant (Mimno and Thompson, 2017). In line with standard SGNS vectors, the frequency of words has relatively little effect on their inner product, with the exception of the rare words, which have slightly less positive inner products. **CBOW** vectors have a relatively large average inner product (4.2985). The vectors trained by **GloVe** show a clear relationship with word frequency, with low-frequency words opposing the frequency-balanced mean vector. The embeddings are well-dispersed, with an average inner product of 0.0002. Finally, **FastText** vectors have a large, positive inner product with the mean (0.2988), indicating that they are not evenly dispersed through the space, but pointing in roughly the same direction. The FastText vectors exhibit a frequency bias, much like what has been previously observed with GloVe vectors. The differences are the results of the inductive biases of the different embedding algorithms.

3 Experiments

This section presents our data, the hyper-parameters of our embeddings, our experimental protocols, and our results.

3.1 Data

In the following experiments we learn word embeddings on samples of a publicly available Wikipedia dump from March 2018.[1] The data is preprocessed using a publicly available pre-processing script[2], extracting text, removing non-alphanumeric characters, converting digits to text, and lowercasing the text.

3.2 Hyper-parameters

We train 300-dimensional word embeddings using the algorithms' recommended hyperparameter settings, listed in the following:[3] For **Hyperwords-SGNS**, the window size is set to 2 and the subsampling of frequent words and smoothing of the context distribution are disabled. The minimal word count for being in the vocabulary is 100. The same applies for **Hyperwords-SVD**, and the exponent for weighting the eigenvalue matrix is 0.5. For **CBOW**, the window size is set to 8, the number of negative samples is 25, and the subsampling threshold for frequent words is 1e-4. For **GloVe**, the window size is set to 15 and the cutoff parameter x$_{max}$ to 10. Finally, for **FastText**, the window size is 5, the number of negatives samples is 5 and the sampling threshold is 0.0001.

3.3 Main experiments

We train word embeddings using the different embedding algorithms listed in §3.2 on two non-overlapping 10% samples of the English Wikipedia dump (the samples contain 463,576 and 528,556 distinct words, with an overlap in vocabulary of 351,858 words). We learn unsupervised and supervised alignments for embeddings (as described in §2) trained by different algorithms on the same datasplits, and for embeddings trained by the same algorithm on the two different datasplits. For the unsupervised alignments, we use the default parameters of the MUSE system for the adversarial training, i.e. a discriminator with 2 fully connected layers of 2048 units trained over 5 epochs, 1,000,000 iterations per epoch with 5 discriminator steps per iteration and a batch size of 32.

We evaluate the alignments in terms of Precision@1 in the word translation retrieval task for the 1500 test words used by Bojanowski et al. (2017). The results are shown in Table 1[4]. Our main observations are: (a) MUSE learns perfect alignments for embeddings learned by the same algorithm on different data splits. (b) MUSE cannot learn alignments for embeddings learned by different algorithms on the same data splits, even if there exists a linear transformation aligning both sets of embeddings (the supervised algorithm learns perfect alignments). We also verify that MUSE cannot learn to align embeddings from different algorithms *even when induced from the same sample*. As already mentioned, we also ran experiments to check that the failure of MUSE to learn good alignments was not a result of the differences in hyper-parameter settings. §3.4 presents additional experiments with normalization, for control; §3.5 addresses how much data is needed to align independently induced embeddings from the same algorithm. §4 discusses potential answers to why MUSE fails when embeddings are induced using different algorithms.

3.4 Experiments with normalization

The embeddings in the main experiments differ in several ways; see §2. One possible explanation for the inability of MUSE to align embeddings from different algorithms could be that the two embeddings are so far apart in space that the discriminator learns to discriminate between them too quickly. Recall that Ω is initialized as the identity matrix I, which means that the generator initially presents the discriminator with the source embedding as is. This is an effect that has often been observed with GANs (Arjovsky and Bottou, 2017); could this also be the explanation for our results? At a first glance, this seems a possible explanation. The inner products with the mean differ significantly for the five embedding

[1] https://dumps.wikimedia.org/enwiki/

[2] http://mattmahoney.net/dc/textdata.html

[3] We also ran experiments with one of the embedding algorithms (FastText) to check if our results were robust across hyper-parameter settings

[4] We report Precision at 1 scores but find that the pattern is the same for Precision at 10, with perfect alignments for embeddings from the same algorithm and 0 scores for alignments between embeddings from different algorithms in the unsupervised experiments.

	Hyperwords-SGNS	Hyperwords-SVD	CBOW	GloVe	FastText
UNSUPERVISED					
Hyperwords-SGNS	**0.997**				
Hyperwords-SVD	0.000	**0.992**			
CBOW	0.000	0.000	**0.997**		
GloVe	0.000	0.000	0.000	**0.997**	
FastText	0.000	0.000	0.000	0.000	**0.997**
SUPERVISED					
Hyperwords-SVD	0.967				
CBOW	0.990	0.989			
GloVe	0.985	0.992	0.999		
FastText	0.994	0.994	0.999	0.997	

Table 1: Precision at 1 (P@1) for unsupervised GAN alignment with Procrustes refinement (top) and supervised Procrustes analysis for the cases in which unsupervised alignment fails (bottom). Results clearly show that GANs can align two independent embeddings induced by the same algorithm; but not embeddings aligned by different ones. Supervised Procrustes analysis, on the other hand, perfectly aligns the embeddings in both cases.

algorithms (see §2). The only embeddings that have roughly the same directionality are Hyperwords and GloVe, and their centroids are very far apart in cosine space. The cosine similarity of the centroids of the two versions of Hyperwords is -0.006, and the cosine similarity for Hyperwords-SVD and GloVe is 0.019. However, poor initialization as a result of applying the identity transform to very distant word embeddings is not the explanation for the poor performance of MUSE in this set-up: Both sets of Hyperwords embeddings were normalized, but alignment still failed. To verify this holds in general, i.e., that results are not affected by normalization in general, we also ran experiments with the remaining 14 embedding pairs, normalizing and/or centering both embeddings. Results stayed the same: Precision at 1 scores of 0.

3.5 Learning curve

MUSE perfectly aligns independently induced word embeddings induced by the same algorithm. For FastText, it correctly aligns 99.7% of all words in the evaluation lexicon with itself. Our samples are 10% of a publicly available Wikipedia dump, amounting to more than 400 million tokens per sample. English-English alignment is an interesting control experiment for unsupervised bilingual dictionary induction, abstracting away from linguistic differences, and we ran a series of experiments to see how small samples MUSE can align in the absence of linguistic differences. The learning curve is presented in Figure 1.

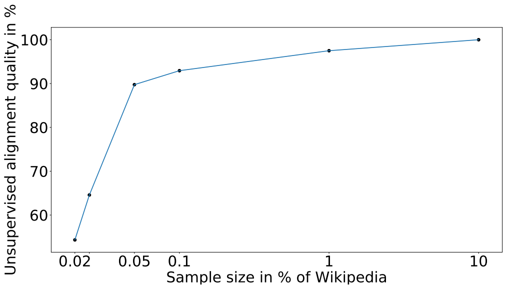

Figure 1: Unsupervised alignment quality for FastText embeddings trained on samples of different sizes, evaluated on 878 words covered by all of the embeddings. The x-axis is log-scaled.

4 Discussion

We have shown that the fact that MUSE cannot align two embedding spaces for English induced by different algorithms (even if using the same corpus), is *not* a result of there not being a linear transformation, and not a result of (lack of) normalization or trivial differences in model hyperparameters. The only explanation left seems to be that the inductive biases of the different algorithms lead to a loss landscape so riddled with local optima that MUSE cannot possible escape them.

To support this hypothesis, compare the loss curves for the MUSE runs aligning embeddings induced with the *same* algorithms (black curves) to the runs aligning embeddings induced with dif-

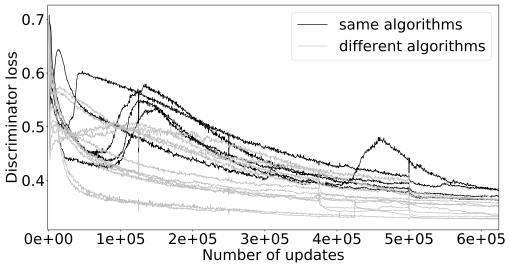

Figure 2: Discriminator losses using the same algorithm for source and target (black curves) or using different algorithms (grey curves).

ferent algorithms, in Figure 2. When the embeddings are induced by the same algorithm, we clearly see the contours of a min-max game, suggesting that the generator and discriminator challenge each other, both contributing to a good alignment. When the embeddings are induced by different algorithms, however, the discriminator quickly drops, with the generator unable to push the discriminator out of a local optimum. *Understanding when biases induce highly non-convex landscapes, and how to make adversarial training less sensitive to such scenarios, remains an open problem, which we think will be key to the success of unsupervised machine translation and related tasks.*

References

Martin Arjovsky and Leon Bottou. 2017. Towards principled methods for training generative adversarial networks. In *ICLR*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of AAAI*.

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.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word Translation Without Parallel Data. In *Proceedings of ICLR*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David WardeFarley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. In *Proceedings of NIPS*.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL* 3:211–225.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*.

Tomas Mikolov, Kai Chen, Gregroy S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of EMNLP*.

Jeff Mitchell and Mark Steedman. 2015. Orthogonality of syntax and semantics within distributional spaces. In *Proceedings of ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.

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

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of ICLR (Conference Track)*.

Anders Søgaard, Sebastian Ruder, and Ivan Vulic. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of ACL*.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of ACL*.