Deconstructing word embedding algorithms

Kian Kenyon-Dean[†][*]
BMO AI Capabilities Team
Bank of Montreal - Toronto, Ontario
kian.kenyon-dean@bmo.com

Edward Newell[*], **Jackie Chi Kit Cheung**
Mila - Québec AI Institute
McGill University - Montréal, Québec
edward.newell@gmail.com
jcheung@cs.mcgill.ca

Abstract

Word embeddings are reliable feature representations of words used to obtain high quality results for various NLP applications. Uncontextualized word embeddings are used in many NLP tasks today, especially in resource-limited settings where high memory capacity and GPUs are not available. Given the historical success of word embeddings in NLP, we propose a retrospective on some of the most well-known word embedding algorithms. In this work, we deconstruct *Word2vec*, *GloVe*, and others, into a common form, unveiling some of the common conditions that seem to be required for making performant word embeddings. We believe that the theoretical findings in this paper can provide a basis for more informed development of future models.

1 Introduction

The advent of efficient uncontextualized word embedding algorithms (e.g., Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)) marked a historical breakthrough in NLP. Countless researchers employed word embeddings in new models to improve results on a multitude of NLP problems. In this work, we provide a retrospective analysis of these groundbreaking models of the past, which simultaneously offers theoretical insights for how future models can be developed and understood. We build on the theoretical work of Levy and Goldberg (2014), proving that their findings on the relationship between pointwise mutual information (PMI) and word embeddings go beyond Word2vec and singular value decomposition.

In particular, we generalize several word embedding algorithms into a common form by proposing the *low rank embedder* framework. We deconstruct each algorithm into its constituent parts, and find that, despite their many different hyperparameters, the algorithms collectively intersect upon the following two key design features. First, vector-covector dot products are learned to approximate PMI statistics in the corpus. Second, modulation of the loss gradient, directly or indirectly, is necessary to balance weak and strong signals arising from the highly imbalanced distribution of corpus statistics.

These findings can provide an informed basis for future development of both new embedding algorithms and deep contextualized models.

2 Fundamental concepts

We begin by formally defining embeddings, their vectors and covectors (also known as “input” and “output” vectors (Rong, 2014; Nalisnick et al., 2016)), and pointwise mutual information (PMI).

Embedding. In general topology, an embedding is understood as an injective structure preserving map, $f : X \to Y$, between two mathematical structures X and Y. A word embedding algorithm (f) learns an inner-product space (Y) to preserve a linguistic structure within a reference corpus of text, $\mathcal{D}$ (X), based on a vocabulary, $\mathcal{V}$. The structure in $\mathcal{D}$ is analyzed in terms of the relationships between words induced by their co-appearances, according to a certain definition of context. In such an analysis, each word figures dually: (1) as a focal element inducing a local context; and (2) as elements of the local contexts induced by focal elements. To make these dual roles explicit, we distinguish two copies of the vocabulary: the *focal*, or *term*, words $\mathcal{V}_T$, and the *context* words $\mathcal{V}_C$.

Word embedding consists of two maps:

$$\begin{aligned} \mathcal{V}_C &\longrightarrow \mathbb{R}^{1\times d} & \qquad \mathcal{V}_T &\longrightarrow \mathbb{R}^{d\times 1} \\ i &\longmapsto \langle i| & j &\longmapsto |j\rangle. \end{aligned}$$

We use Dirac notation to distinguish *vectors* $|j\rangle$, associated to focal words, from *covectors* $\langle i|$, asso-

[*] Kian and Edward contributed equally. † This work was pursued while Kian was a member of Mila.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8479–8484,
November 16–20, 2020. ©2020 Association for Computational Linguistics

ciated to context words. In matrix notation, $|j\rangle$ corresponds to a column vector and $\langle i|$ to a row vector. Their inner product is $\langle i|j\rangle$. We later demonstrate that many word embedding algorithms, intentionally or not, learn a vector space where the inner product between a focal word j and context word i aims to approximate their PMI in the reference corpus: $\langle i|j\rangle \approx \mathrm{PMI}(i,j)$.

Pointwise mutual information (PMI). PMI is a commonly used measure of association in computational linguistics, and has been shown to be consistent and reliable for many tasks (Terra and Clarke, 2003). It measures the deviation of the cooccurrence probability between two words i and j from the product of their marginal probabilities:

$$\mathrm{PMI}(i,j) := \ln \frac{p_{ij}}{p_i p_j} = \ln \frac{N N_{ij}}{N_i N_j}, \tag{1}$$

where p_{ij} is the probability of word i and word j cooccurring (for some notion of cooccurrence), and where p_i and p_j are marginal probabilities of words i and j occurring. The empirical PMI can be found by replacing probabilities with corpus statistics. Words are typically considered to cooccur if they are separated by no more than w words; N_{ij} is the number of counted cooccurrences between a *context* i and a *term* j; N_i, N_j, and N are computed by marginalizing over the N_{ij} statistics.

3 Word embedding algorithms

We will now introduce the *low rank embedder* framework for deconstructing word embedding algorithms, inspired by the theory of generalized low rank models (Udell et al., 2016). We unify several word embedding algorithms by observing them all from the common vantage point of their *global loss function*. Note that this framework is used for theoretical analysis, not necessarily implementation.

The global loss function for a *low rank embedder* takes the following form:

$$\mathcal{L} = \sum_{(i,j)\in\mathcal{V}_C\times\mathcal{V}_T} f_{ij}\Big(\ \psi(\langle i|, |j\rangle),\ \ \phi(i,j)\ \Big), \tag{2}$$

where $\psi(\langle i|, |j\rangle)$ is a kernel function of learned model parameters, and $\phi(i,j)$ is some scalar function (such as a measure of association based on how often i and j appear in the corpus); we denote these with ψ_{ij} and ϕ_{ij} for brevity. As well, f_{ij} are loss functions that take ψ_{ij} and ϕ_{ij} as inputs; all f_{ij} satisfy the property:

$$\frac{\partial f_{ij}}{\partial \psi_{ij}} = 0 \quad \text{at} \quad \psi_{ij} = \phi_{ij}. \tag{3}$$

The design variable ϕ_{ij} is some function of *corpus statistics*, and its purpose is to quantitatively measure some relationship between words i and j. The design variable ψ_{ij} is a function of *model parameters* that aims to approximate ϕ_{ij}; i.e., an embedder's fundamental objective is to learn $\psi_{ij} \approx \phi_{ij}$, and thus to train embeddings that capture the statistical relationships measured by ϕ_{ij}. The simplest choice for the kernel function ψ_{ij}, is to take $\psi_{ij} = \langle i|j\rangle$. But the framework allows any function that is symmetric and positive definite, allowing the inclusion of bias parameters (e.g. in GloVe) and subword parameterization (e.g. in FastText). We later demostrate that skip-gram with negative sampling takes $\phi_{ij} := \mathrm{PMI}(i,j) - \ln k$ and $\psi_{ij} := \langle i|j\rangle$, and then learns parameter values that approximate $\langle i|j\rangle \approx \mathrm{PMI}(i,j) - \ln k$.

To understand the range of models encompassed, it is helpful to see how the framework relates (but is not limited) to matrix factorization. Consider ϕ_{ij} as providing the entries of a matrix: $\mathbf{M} := [\phi_{ij}]_{ij}$. For models that take $\psi_{ij} = \langle i|j\rangle$, we can write $\hat{\mathbf{M}} = \mathbf{WV}$, where $\mathbf{W}$ is defined as having row i equal to $\langle i|$, and $\mathbf{V}$ as having column j equal to $|j\rangle$. Then, the loss function can be rewritten as:

$$\mathcal{L} = \sum_{(i,j)\in\mathcal{V}_C\times\mathcal{V}_T} f_{ij}\Big((\mathbf{WV})_{ij},\ \mathbf{M}_{ij}\Big).$$

This loss function can be interpreted as matrix reconstruction error, because the constraint in Eq. 3 means that the gradient goes to zero as $\mathbf{WV} \approx \mathbf{M}$.

Selecting a particular low rank embedder instance requires key design choices to be made: we must chose the embedding dimension d, the form of the loss terms f_{ij}, the kernel function ψ_{ij}, and the association function ϕ_{ij}. The derivative of f_{ij} with respect to ψ_{ij}, which we call the *characteristic gradient*, helps compare models because it exhibits the action of the gradient yet is symmetric in the parameters. In the Appendix we show how this derivative relates to gradient descent.

In the following subsections, we present the derivations of $\frac{\partial f_{ij}}{\partial \psi_{ij}}$, ψ_{ij}, and ϕ_{ij} for SVD (Levy and Goldberg, 2014; Levy et al., 2015), SGNS (Mikolov et al., 2013), FastText (Joulin et al., 2017), GloVe (Pennington et al., 2014), and LDS (Arora et al., 2016). The derivation for Swivel (Shazeer

Model	$\frac{\partial f_{ij}}{\partial \psi_{ij}}$	ψ_{ij}	ϕ_{ij}	$\langle i\|j\rangle \approx$
SVD	$2 \cdot \big[\psi_{ij} - \phi_{ij}\big]$	$\langle i\|j\rangle$	$\text{PMI}(i,j)$	$\text{PMI}(i,j)$
SGNS	$(N_{ij} + N_{ij}^-) \cdot \big[\sigma(\psi_{ij}) - \sigma(\phi_{ij})\big]$	$\langle i\|j\rangle$	$\ln \frac{N_{ij}}{N_{ij}^-}$	$\text{PMI}(i,j) - \ln k$
GloVe	$2h(N_{ij}) \cdot \big[\psi_{ij} - \phi_{ij}\big]$	$\langle i\|j\rangle + b_i + b_j$	$\ln N_{ij}$	$\text{PMI}(i,j)$
LDS	$4h(N_{ij}) \cdot \Big[\psi_{ij} - \phi_{ij} + C\Big]$	$\\|\langle i\| + \|j\rangle^\intercal\\|^2$	$\ln N_{ij}$	$d\text{PMI}(i,j) - d\gamma$
Swivel	$\sqrt{N_{ij}} \cdot \Big[\psi_{ij} - \phi_{ij}\Big]$	$\langle i\|j\rangle$	$\text{PMI}(i,j)$	$\text{PMI}(i,j)$
	$1 \cdot \sigma\Big(\psi_{ij} - \phi_{ij}\Big)$		$\text{PMI}^*(i,j)$	$\text{PMI}^*(i,j)$

Table 1: Comparison of low rank embedders. Final column shows the value of $\langle i|j\rangle$ at $\frac{\partial f_{ij}}{\partial \psi_{ij}} = 0$. GloVe and LDS set $f_{ij} = 0$ when $N_{ij} = 0$; $h(N_{ij})$ is a weighting function sublinear in N_{ij}. Swivel takes one form when $N_{ij} > 0$ (first row) and another when $N_{ij} = 0$ (second row). N_{ij}^- is the number of negative samples; in SGNS, $N_{ij}^- \propto N_i N_j$, and both N_{ij} and N_{ij}^- are tempered by undersampling and unigram smoothing.

et al., 2016) as a low rank embedder is trivial, as it is already posed as a matrix factorization of PMI statistics. We summarize the derivations in Table 1.

3.1 SVD as a low rank embedder

Singular value decomposition (SVD) of the positive-PMI (PPMI) matrix is used by Levy and Goldberg (2014); Levy et al. (2015) to produce word embeddings that perform more or less equivalently to SGNS and GloVe. Converting the PMI matrix into PPMI is a trivial preprocessing step; ϕ is augmented according to a factor $\alpha = 0$ such that $\phi_{ij} = 0 \; \forall \phi_{ij} \leq \alpha$. We now prove why SVD of the PMI matrix results in word embeddings with dot products $\langle i|j\rangle \approx \text{PMI}(i,j)$, noting that this proof naturally holds for all augmentations of ϕ according to the α factor, including PPMI.

Proof. Truncated SVD provides an optimal solution to problem $\min_D \|D - A\|_F$ for some integer K less than the dimensionality of matrix A such that $\text{rank}(D) = K$ (Udell et al., 2016). The solution is the truncated SVD of A where $D = \sum_{k=1}^{K} \sigma_k u_k v_k^\intercal$ with σ being the k^{th} singular value and u_k and v_k as the k^{th} left and right singular vectors.

Within our framework, the truncated SVD of the PMI matrix thus solves the following loss function (note $A_{ij} = \phi_{ij} = \text{PMI}(i,j)$):

$$\mathcal{L} = -\sum_{(i,j)\in\mathcal{V}_C\times\mathcal{V}_T} \big(\psi_{ij} - \text{PMI}(i,j)\big)^2, \qquad (4)$$

where $\psi_{ij} = u_i^\intercal \Sigma v_j$. Allowing the square matrix of singular values Σ to be absorbed into the vectors (as in Levy et al. (2015)), we have $\langle i| = u_i$ and $|j\rangle = \Sigma v_j$. Thus, taking the derivative $\frac{\partial f_{ij}}{\partial \psi_{ij}}$ (noting that f_{ij} here is simply the squared difference between ψ_{ij} and ϕ_{ij}) and setting it equal to zero we observe:

$$\langle i|j\rangle = \text{PMI}(i,j). \qquad (5)$$

3.2 SGNS as a low rank embedder

Mikolov et al. (2013) proposed skip-gram with negative sampling with the following loss function:

$$\mathcal{L} = -\sum_{(i,j)\in D_2} \Big\{ \ln \sigma\langle i|j\rangle + \sum_{\ell=1}^{k} \mathbb{E}\Big[\ln(1 - \sigma\langle i'_\ell|j\rangle)\Big]\Big\},$$

where σ is the logistic sigmoid function, D_2 is a *list* containing each cooccurrence of a context-word i with a focal word j in the corpus, and the expectation is taken by drawing i'_ℓ from the (smoothed) unigram distribution to generate k "negative samples" for a given focal-word (Mikolov et al., 2013). We will demonstrate that SGNS is a low rank embedder with $\langle i|j\rangle \approx \text{PMI} - \ln k$.

Proof. We can transform the loss function by counting the number of times each pair occurs in the corpus, N_{ij}, and the number of times each pair is drawn as a negative sample, N_{ij}^-, while indexing the sum over the set $\mathcal{V}_C \times \mathcal{V}_T$:

$$\mathcal{L} = -\sum_{(i,j)\in\mathcal{V}_C\times\mathcal{V}_T} \Big\{ N_{ij} \ln \sigma\langle i|j\rangle + N_{ij}^- \ln(1 - \sigma\langle i|j\rangle)\Big\}.$$

The global loss is almost in the required form for a low rank embedder (Eq. 2), and the appropriate setting for the model approximation function is $\psi_{ij} = \langle i|j\rangle$. Calculating the partial derivative with respect to the model approximation function ψ_{ij}, following algebraic manipulation (using the identity $a \equiv (a+b)\sigma(\ln \frac{a}{b})$), we arrive at the following definition of the characteristic gradient for SGNS as a low rank embedder, where $\frac{\partial f_{ij}}{\partial \psi_{ij}} = \frac{\partial \mathcal{L}}{\partial \langle i|j\rangle}$:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \langle i|j\rangle} &= N^{-}_{ij}\sigma\langle i|j\rangle - N_{ij}(1-\sigma\langle i|j\rangle) \\
&= (N_{ij} + N^{-}_{ij})\Big[\sigma\big(\langle i|j\rangle\big) - \sigma\Big(\ln\frac{N_{ij}}{N^{-}_{ij}}\Big)\Big] \\
&= (N_{ij} + N^{-}_{ij})\Big[\sigma\big(\psi_{ij}\big) - \sigma\big(\phi_{ij}\big)\Big]. \qquad (6)
\end{aligned}
$$

This provides that the association function for SGNS is $\phi_{ij} = \ln(N_{ij}/N^{-}_{ij})$, since the derivative will be equal to zero at that point (Eq. 3). However, recall that negative samples are drawn according to the unigram distribution (or a smoothed variant (Levy et al., 2015)). This means that $N^{-}_{ij} = kN_iN_j/N$. Therefore, in agreement with Levy and Goldberg (2014), we find that:

$$\phi_{ij} = \ln\frac{N_{ij}N}{N_iN_jk} = \mathrm{PMI}(i,j) - \ln k. \qquad (7)$$

3.3 FastText as a low rank embedder

Proposed by Joulin et al. (2017), FastText's motivation is orthogonal to the present work. Its purpose is to provide subword-based representation of words to improve vocabulary coverage and generalizability of word embeddings. Nonetheless, it can also be understood as a low rank embedder .

Proof. FastText uses a loss function that is identical to SGNS except that the vector for each word is taken as the sum of embeddings for all character n-grams appearing in the word, with $3 \le n \le 6$. Therefore, define $|j\rangle$ by $|j\rangle \equiv \sum_{g\in z(j)} |g\rangle$, where $|g\rangle$ is the vector for n-gram g, and $z(j)$ is the set of n-grams in word j. Covectors are accorded to words directly, so need not be redefined. The loss function and the derivation of entries for Table 1 is then formally identical to those for SGNS. This provides that $\psi_{ij} = \langle i|j\rangle$, and, $\phi_{ij} = \mathrm{PMI}(i,j) - \ln k$.

3.4 GloVe as a low rank embedder

GloVe was proposed as an algorithm halfway between sampling methods and matrix factorization (Pennington et al., 2014). Ignoring samples where $N_{ij} = 0$, GloVe uses the following loss function:

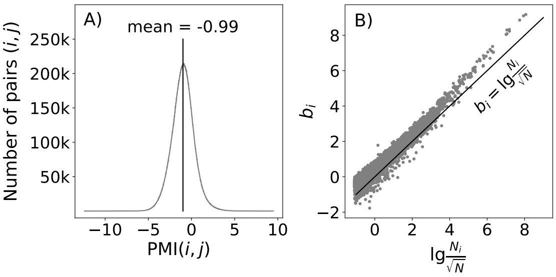

Figure 1: **A**) Histogram of $\mathrm{PMI}(i,j)$ values, for all pairs (i,j) with $N_{ij} > 0$. **B**) Scatter plot of GloVe's learned biases. Both from a Wikipedia 2018 corpus.

$$\mathcal{L} = \sum_{ij} h(N_{ij})\Big(\langle i|j\rangle + b_i + b_j - \ln N_{ij}\Big)^2 \qquad (8)$$

where b_i and b_j are learned bias parameters, and $h(N_{ij})$ is a weighting function sublinear in N_{ij}.

GloVe can be cast as a low rank embedder by using the model approximation function as a kernel with bias parameters, and setting the association measure to simply be the objective:

$$
\begin{aligned}
\psi_{ij} &= \big[\,\langle i|_1 \cdots \langle i|_d\, b_i\; 1\,\big] \cdot \big[\,|j\rangle_1 \cdots |j\rangle_d\;\; 1\;\; b_j\,\big]^{\intercal}, \\
&\text{and} \quad \phi_{ij} = \ln N_{ij}.
\end{aligned}
$$

Proof. Observe an optimal solution to the loss function, when $\frac{\partial f_{ij}}{\partial \psi_{ij}} = 0$:

$$
\begin{aligned}
\frac{\partial f_{ij}}{\partial \psi_{ij}} &= 2h(N_{ij})\Big[\langle i|j\rangle + b_i + b_j - \ln N_{ij}\Big] = 0 \\
&\implies \langle i|j\rangle + b_i + b_j = \ln N_{ij}.
\end{aligned}
$$

Multiplying the log operand by 1:

$$
\begin{aligned}
\langle i|j\rangle + b_i + b_j &= \ln\left(\frac{N_iN_j}{N}\frac{N}{N_iN_j}N_{ij}\right) \qquad (9) \\
&= \ln\frac{N_i}{\sqrt{N}} + \ln\frac{N_j}{\sqrt{N}} + \mathrm{PMI}(i,j). \\
&\qquad (10)
\end{aligned}
$$

On the right side, we have two terms that depend respectively only on i and j, which are candidates for the bias terms. Based on this equation alone, we cannot draw any conclusions. However, empirically the bias terms are in fact very near $\frac{N_i}{\sqrt{N}}$ and $\frac{N_j}{\sqrt{N}}$, and PMI is observed to be normally distributed, as can be seen in Fig. 1. This means that Eq. 10 provides $\langle i|j\rangle \approx \mathrm{PMI}(i,j)$.

Analyzing the optimum of GloVe's loss function yields important insights. First, GloVe can be added to the list of low rank embedders that learn a bilinear parameterization of PMI. Second, we can see why such a parameterization is advantageous. Generally, it helps to standardize features of low rank models (Udell et al., 2016), and this is essentially what transforming cooccurrence counts into PMI achieves. Thus, PMI can be viewed as a parameterization trick, providing an approximately normal target association to be modelled.

3.5 LDS as a low rank embedder

Arora et al. (2016) introduced an embedding perspective based on generative modelling with random walks through a latent discourse space (LDS). LDS provided a theoretical basis for the performant *SIF document embedding* algorithm, developed soon afterwards (Arora et al., 2017). We now demonstrate that LDS is also a low-rank embedder.

Proof. The low rank learning objective for LDS follows directly from **Corollary 2.3**, in Arora et al. (2016):

$$\mathrm{PMI}(i,j) = \frac{\langle i|j\rangle}{d} + \gamma + O(\epsilon).$$

$\frac{\partial f_{ij}}{\partial \psi_{ij}}$ can be found by straightforward differentiation of LDS's loss function:

$$\mathcal{L} = \sum_{ij} h(N_{ij}) \big[\ln N_{ij} - \| \langle i| + |j\rangle^\intercal \|^2 - C \big]^2,$$

where $h(N_{ij})$ is as defined by GloVe. The quadratic term is a valid kernel function because:

$$\frac{\partial f_{ij}}{\partial \psi_{ij}} = \| \langle i| + |j\rangle^\intercal \|^2 = \langle \tilde{i} | \tilde{j} \rangle,$$

where

$$\begin{aligned} \langle \tilde{i}| &= \Big[\sqrt{2}\langle i|_1 \;\cdots\; \sqrt{2}\langle i|_d \;\; \langle i|\langle i|^\intercal \;\;\; 1 \;\;\; \Big], \\ |\tilde{j}\rangle &= \Big[\sqrt{2}|j\rangle_1 \;\cdots\; \sqrt{2}|j\rangle_d \;\;\; 1 \;\;\; |j\rangle^\intercal|j\rangle \;\Big]^\intercal. \end{aligned}$$

4 Related work

Our derivation of SGNS's solution is inspired by the work of Levy and Goldberg (2014), who proved that *skip-gram with negative sampling* (SGNS) (Mikolov et al., 2013) was implicitly factorizing the PMI $-$ $\ln k$ matrix. However, they required additional assumptions for their derivation to hold. Li et al. (2015) explored relations between SGNS and matrix factorization, but their derivation diverges from Levy and Goldberg's result and masks the connection between SGNS and other low rank embedders. Other works have also explored theoretical or empirical relationships between SGNS and GloVe (Shi and Liu, 2014; Suzuki and Nagata, 2015; Levy et al., 2015; Arora et al., 2016).

5 Discussion

We observe common features between each of the algorithms (Table 1). In each case, $\frac{\partial f_{ij}}{\partial \psi_{ij}}$ takes the form $(\text{multiplier}) \cdot (\text{difference})$. The multiplier is always a "tempered" version of N_{ij} (or $N_i N_j$); that is, it increases sublinearly with N_{ij}.

For each algorithm, ϕ_{ij} is equal to PMI or a scaled log of N_{ij}. Yet, the choice of ψ_{ij} in combination with ϕ_{ij} provides that every model is optimized when $\langle i|j\rangle$ tends toward $\mathrm{PMI}(i,j)$ (with or without a constant shift or scaling). We demonstrated that the optimum for SGNS (and FastTest) is equivalent to the shifted PMI (§3.2). For GloVe, we showed that incorporation of the bias terms captures the unigram counts needed for PMI (§3.4). A similar property is found in LDS with regards to the L2 norm in its learning objective (Arora et al., 2016). Thus, these algorithms all converge on two key points: (1) an optimum in which model parameters are bilinearly related to PMI; and, (2) the weighting of $\frac{\partial f_{ij}}{\partial \psi_{ij}}$ by some tempered form of N_{ij}.

6 Conclusion

Our *low rank embedder* framework has evoked the commonalities between many word embedding algorithms. We believe a robust understanding of these algorithms is a prerequisite for theoretically motivated development of deeper models. Indeed, we offer the following conjectures: deep embedding models would benefit by incorporating PMI statistics into their training objective; such models will also benefit from sub-linear scaling of frequent word pairs during training; and, lastly, such models would benefit by learning representations with a dual character, as all of the embedding algorithms we described do by learning vectors *and* covectors.

Acknowledgements

This work is supported by the Fonds de recherche du Québec – Nature et technologies and by the Natural Sciences and Engineering Research Council of Canada. The last author is supported in part by the Canada CIFAR AI Chair program.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. *International Conference on Learning Representations*.

Armand Joulin, Edouard Grave, and Piotr Bojanowski Tomas Mikolov. 2017. Bag of tricks for efficient text classification. *European Association for Computational Linguistics 2017*, page 427.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong, and Enhong Chen. 2015. Word embedding revisited: a new representation learning and explicit matrix factorization perspective. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3650–3656. AAAI Press.

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

Eric Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. 2016. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 83–84. International World Wide Web Conferences Steering Committee.

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*, pages 1532–1543.

Xin Rong. 2014. Word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. 2016. Swivel: Improving embeddings by noticing what's missing. *arXiv preprint arXiv:1602.02215*.

Tianze Shi and Zhiyuan Liu. 2014. Linking glove with word2vec. *arXiv preprint arXiv:1411.5595*.

Jun Suzuki and Masaaki Nagata. 2015. A unified learning framework of skip-grams and global vectors. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP (Volume 2: Short Papers)*, volume 2, pages 186–191.

Egidio Terra and Charles LA Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Proceedings of the 2003 Conference of NAACL-HLT - Volume 1*, pages 165–172. Association for Computational Linguistics.

Madeleine Udell, Corinne Horn, Reza Zadeh, Stephen Boyd, et al. 2016. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118.

A Appendix

A.1 On the characteristic gradient

The relationship between $\frac{\partial f_{ij}}{\partial \psi_{ij}}$ and the gradient descent actions taken during learning requires simply taking the next step in the chain rule during differentiation. For simplicity of exposition, we will assume, like SGNS and Swivel, that $\psi_{ij} = \langle i|j\rangle$, although the motivation of taking this derivative holds for any definition of ψ_{ij}, provided that it is a kernel function of the model parameters.

By examining the derivative $\frac{\partial f_{ij}}{\partial \langle i|j\rangle}$ we observe the primary objective of the model (to approximate dot products), and how this objective symmetrically updates vectors and covectors during learning.

Consider the generic update that occurs for a single (i, j) pair with the pairwise loss function f_{ij}. The gradient descent rule for a single update to the vector for word j, using some learning rate η, is:

$$|j\rangle \leftarrow |j\rangle - \eta \frac{\partial f_{ij}}{|j\rangle}, \tag{11}$$

However, since f_{ij} is a function of $\langle i|j\rangle$ and not of the vectors or covectors independently, we can use the chain rule to arrive at the following:

$$|j\rangle \leftarrow |j\rangle - \eta \frac{\partial f_{ij}}{\partial \langle i|j\rangle} \frac{\partial \langle i|j\rangle}{\partial |j\rangle} \tag{12}$$

$$|j\rangle \leftarrow |j\rangle - \eta \frac{\partial f_{ij}}{\partial \langle i|j\rangle} \langle i|^{\intercal}, \tag{13}$$

since $\frac{\partial \langle i|j\rangle}{\partial |j\rangle} = \langle i|$. Symmetrically, we also arrive at, for the updates to covectors:

$$\langle i| \leftarrow \langle i| - \eta \frac{\partial f_{ij}}{\partial \langle i|j\rangle} |j\rangle^{\intercal}. \tag{14}$$

Therefore, taking $\frac{\partial f_{ij}}{\partial \langle i|j\rangle}$ (more generally, $\frac{\partial f_{ij}}{\partial \psi_{ij}}$) to be the focal point of analysis in determining the objectives of the low rank embedders is well grounded.