A Latent Variable Model Approach to PMI-based Word Embeddings

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, Andrej Risteski
Computer Science Department, Princeton University
35 Olden St, Princeton, NJ 08540
{arora,yuanzhil,yingyul,tengyu,risteski}@cs.princeton.edu

Abstract

Semantic word embeddings represent the meaning of a word via a vector, and are created by diverse methods. Many use nonlinear operations on co-occurrence statistics, and have hand-tuned hyperparameters and reweighting methods.

This paper proposes a new generative model, a dynamic version of the log-linear topic model of Mnih and Hinton (2007). The methodological novelty is to use the prior to compute closed form expressions for word statistics. This provides a theoretical justification for nonlinear models like PMI, word2vec, and GloVe, as well as some hyperparameter choices. It also helps explain why low-dimensional semantic embeddings contain linear algebraic structure that allows solution of word analogies, as shown by Mikolov et al. (2013a) and many subsequent papers.

Experimental support is provided for the generative model assumptions, the most important of which is that latent word vectors are fairly uniformly dispersed in space.

1 Introduction

Vector representations of words (word embeddings) try to capture relationships between words as distance or angle, and have many applications in computational linguistics and machine learning. They are constructed by various models whose unifying philosophy is that the meaning of a word is defined by "the company it keeps" (Firth, 1957), namely, co-occurrence statistics. The simplest methods use word vectors that explicitly represent co-occurrence statistics. Reweighting heuristics are known to improve these methods, as is dimension reduction (Deerwester et al., 1990). Some reweighting methods are nonlinear, which include taking the square root of co-occurrence counts (Rohde et al., 2006), or the logarithm, or the related Pointwise Mutual Information (PMI) (Church and Hanks, 1990). These are collectively referred to as Vector Space Models, surveyed in (Turney and Pantel, 2010).

Neural network language models (Rumelhart et al., 1986; Rumelhart et al., 1988; Bengio et al., 2006; Collobert and Weston, 2008a) propose another way to construct embeddings: the word vector is simply the neural network's internal representation for the word. This method is nonlinear and nonconvex. It was popularized via word2vec, a family of energy-based models in (Mikolov et al., 2013b; Mikolov et al., 2013c), followed by a matrix factorization approach called GloVe (Pennington et al., 2014). The first paper also showed how to solve analogies using linear algebra on word embeddings. Experiments and theory were used to suggest that these newer methods are related to the older PMI-based models, but with new hyperparameters and/or term reweighting methods (Levy and Goldberg, 2014b).

But note that even the old PMI method is a bit mysterious. The simplest version considers a symmetric matrix with each row/column indexed by a word. The entry for (w, w') is $\text{PMI}(w, w') = \log \frac{p(w,w')}{p(w)p(w')}$, where $p(w, w')$ is the empirical probability of words w, w' appearing within a window of certain size in the corpus, and $p(w)$ is the marginal

Transactions of the Association for Computational Linguistics, vol. 4, pp. 385–399, 2016. Action Editor: Daichi Mochihashi.
Submission batch: 10/2015; Revision batch: 2/2016; 3/2016; Published 7/2016.

probability of w. (More complicated models could use asymmetric matrices with columns corresponding to context words or phrases, and also involve tensorization.) Then word vectors are obtained by low-rank SVD on this matrix, or a related matrix with term reweightings. In particular, the PMI matrix is found to be closely approximated by a low rank matrix: there exist word vectors in say 300 dimensions, which is much smaller than the number of words in the dictionary, such that

$$\langle v_w, v_{w'} \rangle \approx \text{PMI}(w, w') \tag{1.1}$$

where $\approx$ should be interpreted loosely.

There appears to be no theoretical explanation for this empirical finding about the approximate low rank of the PMI matrix. The current paper addresses this. Specifically, we propose a probabilistic model of text generation that augments the log-linear topic model of Mnih and Hinton (2007) with dynamics, in the form of a random walk over a latent discourse space. The chief methodological contribution is using the model priors to analytically derive a closed-form expression that directly explains (1.1); see Theorem 2.2 in Section 2. Section 3 builds on this insight to give a rigorous justification for models such as word2vec and GloVe, including the hyperparameter choices for the latter. The insight also leads to a mathematical explanation for why these word embeddings allow analogies to be solved using linear algebra; see Section 4. Section 5 shows good empirical fit to this model's assumtions and predictions, including the surprising one that word vectors are pretty uniformly distributed (isotropic) in space.

1.1 Related work

Latent variable probabilistic models of language have been used for word embeddings before, including Latent Dirichlet Allocation (LDA) and its more complicated variants (see the survey (Blei, 2012)), and some neurally inspired nonlinear models (Mnih and Hinton, 2007; Maas et al., 2011). In fact, LDA evolved out of efforts in the 1990s to provide a generative model that "explains" the success of older vector space methods like Latent Semantic Indexing (Papadimitriou et al., 1998; Hofmann, 1999). However, none of these earlier generative models has been linked to PMI models.

Levy and Goldberg (2014b) tried to relate word2vec to PMI models. They showed that if there were no dimension constraint in word2vec, specifically, the "skip-gram with negative sampling (SGNS)" version of the model, then its solutions would satisfy (1.1), provided the right hand side were replaced by $\text{PMI}(w, w') - \beta$ for some scalar β. However, skip-gram is a discriminative model (due to the use of negative sampling), not generative. Furthermore, their argument only applies to very high-dimensional word embeddings, and thus does not address low-dimensional embeddings, which have superior quality in applications.

Hashimoto et al. (2016) focuses on issues similar to our paper. They model text generation as a random walk on words, which are assumed to be embedded as vectors in a geometric space. Given that the last word produced was w, the probability that the next word is w' is assumed to be given by $h(|v_w - v_{w'}|^2)$ for a suitable function h, and this model leads to an explanation of (1.1). By contrast our random walk involves a latent discourse vector, which has a clearer semantic interpretation and has proven useful in subsequent work, e.g. understanding structure of word embeddings for polysemous words Arora et al. (2016). Also our work clarifies some weighting and bias terms in the training objectives of previous methods (Section 3) and also the phenomenon discussed in the next paragraph.

Researchers have tried to understand why vectors obtained from the highly nonlinear word2vec models exhibit linear structures (Levy and Goldberg, 2014a; Pennington et al., 2014). Specifically, for analogies like "*man*:*woman*::*king*:??," *queen* happens to be the word whose vector v_{queen} is the most similar to the vector $v_{king} - v_{man} + v_{woman}$. This suggests that simple semantic relationships, such as *masculine* vs *feminine* tested in the above example, correspond approximately to a single direction in space, a phenomenon we will henceforth refer to as RELATIONS=LINES.

Section 4 surveys earlier attempts to explain this phenomenon and their shortcoming, namely, that they ignore the large approximation error in relationships like (1.1). This error appears larger than the difference between the best solution and the second best (incorrect) solution in analogy solving, so that this error could in principle lead to a complete

failure in analogy solving. In our explanation, the low dimensionality of the word vectors plays a key role. This can also be seen as a theoretical explanation of the old observation that dimension reduction improves the quality of word embeddings for various tasks. The intuitive explanation often given —that smaller models generalize better—turns out to be fallacious, since the training method for creating embeddings makes no reference to analogy solving. Thus there is no a priori reason why low-dimensional model parameters (i.e., lower model capacity) should lead to better performance in analogy solving, just as there is no reason they are better at some other unrelated task like predicting the weather.

1.2 Benefits of generative approaches

In addition to giving some form of "unification" of existing methods, our generative model also brings more intepretability to word embeddings beyond traditional cosine similarity and even analogy solving. For example, it led to an understanding of how the different senses of a polysemous word (e.g., *bank*) reside in linear superposition within the word embedding (Arora et al., 2016). Such insight into embeddings may prove useful in the numerous settings in NLP and neuroscience where they are used.

Another new explanatory feature of our model is that low dimensionality of word embeddings plays a key theoretical role —unlike in previous papers where the model is agnostic about the dimension of the embeddings, and the superiority of low-dimensional embeddings is an empirical finding (starting with Deerwester et al. (1990)). Specifically, our theoretical analysis makes the key assumption that the set of all word vectors (which are latent variables of the generative model) are spatially isotropic, which means that they have no preferred direction in space. Having n vectors be isotropic in d dimensions requires $d \ll n$. This isotropy is needed in the calculations (i.e., multidimensional integral) that yield (1.1). It also holds empirically for our word vectors, as shown in Section 5.

The isotropy of low-dimensional word vectors also plays a key role in our explanation of the RELATIONS=LINES phenomenon (Section 4). The isotropy has a "purification" effect that mitigates the effect of the (rather large) approximation error in the PMI models.

2 Generative model and its properties

The model treats corpus generation as a dynamic process, where the t-th word is produced at step t. The process is driven by the random walk of a discourse vector $c_t \in \Re^d$. Its coordinates represent what is being talked about.[1] Each word has a (time-invariant) latent vector $v_w \in \Re^d$ that captures its correlations with the discourse vector. We model this bias with a log-linear word production model:

$$\Pr[w \text{ emitted at time } t \mid c_t] \propto \exp(\langle c_t, v_w \rangle). \quad (2.1)$$

The discourse vector c_t does a slow random walk (meaning that c_{t+1} is obtained from c_t by adding a small random displacement vector), so that nearby words are generated under similar discourses. We are interested in the probabilities that word pairs co-occur near each other, so occasional big jumps in the random walk are allowed because they have negligible effect on these probabilities.

A similar log-linear model appears in Mnih and Hinton (2007) but without the random walk. The linear chain CRF of Collobert and Weston (2008b) is more general. The dynamic topic model of Blei and Lafferty (2006) utilizes topic dynamics, but with a linear word production model. Belanger and Kakade (2015) have proposed a dynamic model for text using Kalman Filters, where the sequence of words is generated from Gaussian linear dynamical systems, rather than the log-linear model in our case.

The novelty here over such past works is a theoretical analysis in the method-of-moments tradition (Hsu et al., 2012; Cohen et al., 2012). Assuming a prior on the random walk we analytically integrate out the hidden random variables and compute a simple closed form expression that approximately connects the model parameters to the observable joint probabilities (see Theorem 2.2). This is reminiscent of analysis of similar random walk models in finance (Black and Scholes, 1973).

Model details. Let n denote the number of words and d denote the dimension of the discourse space, where $1 \leq d \leq n$. Inspecting (2.1) suggests word

[1] This is a different interpretation of the term "discourse" compared to some other settings in computational linguistics.

vectors need to have varying lengths, to fit the empirical finding that word probabilities satisfy a power law. Furthermore, we will assume that in the bulk, the word vectors are distributed uniformly in space, earlier referred to as isotropy. This can be quantified as a prior in the Bayesian tradition. More precisely, the ensemble of word vectors consists of i.i.d draws generated by $v = s \cdot \hat{v}$, where $\hat{v}$ is from the spherical Gaussian distribution, and s is a scalar random variable. We assume s is a random scalar with expectation $\tau = \Theta(1)$ and s is always upper bounded by κ, which is another constant. Here τ governs the expected magnitude of $\langle v, c_t \rangle$, and it is particularly important to choose it to be $\Theta(1)$ so that the distribution $\Pr[w|c_t] \propto \exp(\langle v_w, c_t \rangle)$ is interesting.[2] Moreover, the dynamic range of word probabilities will roughly equal $\exp(\kappa^2)$, so one should think of κ as an absolute constant like 5. These details about s are important for realistic modeling but not too important in our analysis. (Furthermore, readers uncomfortable with this simplistic Bayesian prior should look at Section 2.1 below.)

Finally, we clarify the nature of the random walk. We assume that the stationary distribution of the random walk is uniform over the unit sphere, denoted by $\mathcal{C}$. The transition kernel of the random walk can be in any form so long as at each step the movement of the discourse vector is at most $\epsilon_2/\sqrt{d}$ in ℓ_2 norm.[3] This is still fast enough to let the walk mix quickly in the space.

The following lemma (whose proof appears in the appendix) is central to the analysis. It says that under the Bayesian prior, the partition function $Z_c = \sum_w \exp(\langle v_w, c \rangle)$, which is the implied normalization in equation (2.1), is close to some constant Z for most of the discourses c. This can be seen as a plausible theoretical explanation of a phenomenon called *self-normalization* in log-linear models: ignoring the partition function or treating it as a constant (which greatly simplifies training) is known to often give good results. This has also been studied in (Andreas and Klein, 2014).

Lemma 2.1 (Concentration of partition functions)**.** *If the word vectors satisfy the Bayesian prior described in the model details, then*

$$\Pr_{c \sim \mathcal{C}} [(1 - \epsilon_z)Z \leq Z_c \leq (1 + \epsilon_z)Z] \geq 1 - \delta, \quad (2.2)$$

for $\epsilon_z = \widetilde{O}(1/\sqrt{n})$*, and* $\delta = \exp(-\Omega(\log^2 n))$.

The concentration of the partition functions then leads to our main theorem (the proof is in the appendix). The theorem gives simple closed form approximations for $p(w)$, the probability of word w in the corpus, and $p(w, w')$, the probability that two words w, w' occur next to each other. The theorem states the result for the window size $q = 2$, but the same analysis works for pairs that appear in a small window, say of size 10, as stated in Corollary 2.3. Recall that $\text{PMI}(w, w') = \log[p(w, w')/(p(w)p(w'))]$.

Theorem 2.2. *Suppose the word vectors satisfy the inequality* (2.2)*, and window size* $q = 2$*. Then,*

$$\log p(w, w') = \frac{\|v_w + v_{w'}\|_2^2}{2d} - 2 \log Z \pm \epsilon, \quad (2.3)$$

$$\log p(w) = \frac{\|v_w\|_2^2}{2d} - \log Z \pm \epsilon. \quad (2.4)$$

for $\epsilon = O(\epsilon_z) + \widetilde{O}(1/d) + O(\epsilon_2)$*. Jointly these imply:*

$$\textit{PMI}\ (w, w') = \frac{\langle v_w, v_{w'} \rangle}{d} \pm O(\epsilon). \quad (2.5)$$

Remarks 1. Since the word vectors have ℓ_2 norm of the order of $\sqrt{d}$, for two typical word vectors $v_w, v_{w'}$, $\|v_w + v_{w'}\|_2^2$ is of the order of $\Theta(d)$. Therefore the noise level ϵ is very small compared to the leading term $\frac{1}{2d}\|v_w + v_{w'}\|_2^2$. For PMI however, the noise level $O(\epsilon)$ could be comparable to the leading term, and empirically we also find higher error here.

Remarks 2. Variants of the expression for joint probability in (2.3) had been hypothesized based upon empirical evidence in Mikolov et al. (2013b) and also Globerson et al. (2007), and Maron et al. (2010) .

Remarks 3. Theorem 2.2 directly leads to the extension to a general window size q as follows:

[2] A larger τ will make $\Pr[w|c_t]$ too peaked and a smaller one will make it too uniform.

[3] More precisely, the proof extends to any symmetric product stationary distribution $\mathcal{C}$ with sub-Gaussian coordinate satisfying $\mathbb{E}_c\left[\|c\|^2\right] = 1$, and the steps are such that for all c_t, $\mathbb{E}_{p(c_{t+1}|c_t)}[\exp(\kappa\sqrt{d}\|c_{t+1} - c_t\|)] \leq 1 + \epsilon_2$ for some small ϵ_2.

Corollary 2.3. *Let $p_q(w, w')$ be the co-occurrence probability in windows of size q, and $PMI_q(w, w')$ be the corresponding PMI value. Then*

$$\log p_q(w, w') = \frac{\|v_w + v_{w'}\|_2^2}{2d} - 2\log Z + \gamma \pm \epsilon,$$

$$PMI_q\ (w, w') = \frac{\langle v_w, v_{w'}\rangle}{d} + \gamma \pm O(\epsilon).$$

where $\gamma = \log\left(\frac{q(q-1)}{2}\right)$.

It is quite easy to see that Theorem 2.2 implies the Corollary 2.3, as when the window size is q the pair w, w' could appear in any of $\binom{q}{2}$ positions within the window, and the joint probability of w, w' is roughly the same for any positions because the discourse vector changes slowly. (Of course, the error term gets worse as we consider larger window sizes, although for any constant size, the statement of the theorem is correct.) This is also consistent with the shift β for fitting PMI in (Levy and Goldberg, 2014b), which showed that without dimension constraints, the solution to skip-gram with negative sampling satisfies PMI $(w, w') - \beta = \langle v_w, v_{w'}\rangle$ for a constant β that is related to the negative sampling in the optimization. Our result justifies via a generative model why this should be satisfied even for low dimensional word vectors.

2.1 Weakening the model assumptions

For readers uncomfortable with Bayesian priors, we can replace our assumptions with concrete properties of word vectors that are empirically verifiable (Section 5.1) for our final word vectors, and in fact also for word vectors computed using other recent methods.

The word meanings are assumed to be represented by some "ground truth" vectors, which the experimenter is trying to recover. These ground truth vectors are assumed to be spatially isotropic in the bulk, in the following two specific ways: (i) For almost all unit vectors c the sum $\sum_w \exp(\langle v_w, c\rangle)$ is close to a constant Z; (ii) Singular values of the matrix of word vectors satisfy properties similar to those of random matrices, as formalized in the paragraph before Theorem 4.1. Our Bayesian prior on the word vectors happens to imply that these two conditions hold with high probability. But the conditions may hold even if the prior doesn't hold. Furthermore, they are compatible with all sorts of local structure among word vectors such as existence of clusterings, which would be absent in truly random vectors drawn from our prior.

3 Training objective and relationship to other models

To get a training objective out of Theorem 2.2, we reason as follows. Let $X_{w,w'}$ be the number of times words w and w' co-occur within the same window in the corpus. The probability $p(w, w')$ of such a co-occurrence at any particular time is given by (2.3). Successive samples from a random walk are not independent. But if the random walk mixes fairly quickly (the mixing time is related to the *logarithm* of the vocabulary size), then the distribution of $X_{w,w'}$'s is very close to a multinomial distribution $\mathsf{Mul}(\tilde{L}, \{p(w, w')\})$, where $\tilde{L} = \sum_{w,w'} X_{w,w'}$ is the total number of word pairs.

Assuming this approximation, we show below that the maximum likelihood values for the word vectors correspond to the following optimization,

$$\min_{\{v_w\},C} \sum_{w,w'} X_{w,w'} \left(\log(X_{w,w'}) - \|v_w + v_{w'}\|_2^2 - C\right)^2$$

As is usual, empirical performance is improved by weighting down very frequent word pairs, possibly because very frequent words such as "*the*" do not fit our model. This is done by replacing the weighting $X_{w,w'}$ by its truncation $\min\{X_{w,w'}, X_{\max}\}$ where $X_{\max}$ is a constant such as 100. We call this objective with the truncated weights **SN** (**S**quared **N**orm).

We now give its derivation. Maximizing the likelihood of $\{X_{w,w'}\}$ is equivalent to maximizing

$$\ell = \log\left(\prod_{(w,w')} p(w, w')^{X_{w,w'}}\right).$$

Denote the logarithm of the ratio between the expected count and the empirical count as

$$\Delta_{w,w'} = \log\left(\frac{\tilde{L}p(w, w')}{X_{w,w'}}\right). \tag{3.1}$$

Then with some calculation, we obtain the following where c is independent of the empirical observations

$X_{w,w'}$'s.

$$\ell = c + \sum_{(w,w')} X_{w,w'}\Delta_{w,w'} \tag{3.2}$$

On the other hand, using $e^x \approx 1+x+x^2/2$ when x is small,[4] we have

$$\tilde{L} = \sum_{(w,w')} \tilde{L}p_{w,w'} = \sum_{(w,w')} X_{w,w'}e^{\Delta_{w,w'}}$$

$$\approx \sum_{(w,w')} X_{w,w'}\left(1+\Delta_{w,w'}+\frac{\Delta^2_{w,w'}}{2}\right).$$

Note that $\tilde{L} = \sum_{(w,w')} X_{w,w'}$, so

$$\sum_{(w,w')} X_{w,w'}\Delta_{w,w'} \approx -\frac{1}{2}\sum_{(w,w')} X_{w,w'}\Delta^2_{w,w'}.$$

Plugging this into (3.2) leads to

$$2(c-\ell) \approx \sum_{(w,w')} X_{w,w'}\Delta^2_{w,w'}. \tag{3.3}$$

So maximizing the likelihood is approximately equivalent to minimizing the right hand side, which (by examining (3.1)) leads to our objective.

Objective for training with PMI. A similar objective **PMI** can be obtained from (2.5), by computing an approximate MLE, using the fact that the error between the empirical and true value of PMI(w,w') is driven by the smaller term $p(w,w')$, and not the larger terms $p(w), p(w')$.

$$\min_{\{v_w\},C} \sum_{w,w'} X_{w,w'}\left(\text{PMI}(w,w') - \langle v_w, v_{w'}\rangle\right)^2$$

This is of course very analogous to classical VSM methods, with a novel reweighting method.

Fitting to either of the objectives involves solving a version of *Weighted SVD* which is NP-hard, but empirically seems solvable in our setting via AdaGrad (Duchi et al., 2011).

[4]This Taylor series approximation has an error of the order of x^3, but ignoring it can be theoretically justified as follows. For a large $X_{w,w'}$, its value approaches its expectation and thus the corresponding $\Delta_{w,w'}$ is close to 0 and thus ignoring $\Delta^3_{w,w'}$ is well justified. The terms where $\Delta_{w,w'}$ is significant correspond to $X_{w,w'}$'s that are small. But empirically, $X_{w,w'}$'s obey a power law distribution (see, e.g. Pennington et al. (2014)) using which it can be shown that these terms contribute a small fraction of the final objective (3.3). So we can safely ignore the errors. Full details appear in the ArXiv version of this paper (Arora et al., 2015).

Connection to GloVe. Compare **SN** with the objective used by GloVe (Pennington et al., 2014):

$$\sum_{w,w'} f(X_{w,w'})(\log(X_{w,w'}) - \langle v_w, v_{w'}\rangle - s_w - s_{w'} - C)^2$$

with $f(X_{w,w'}) = \min\{X^{3/4}_{w,w'}, 100\}$. Their weighting methods and the need for *bias* terms $s_w, s_{w'}, C$ were derived by trial and error; here they are all predicted and given meanings due to Theorem 2.2, specifically $s_w = \|v_w\|^2$.

Connection to word2vec(CBOW). The CBOW model in word2vec posits that the probability of a word w_{k+1} as a function of the previous k words $w_1, w_2, \ldots, w_k$:

$$p\left(w_{k+1}\middle|\, \{w_i\}_{i=1}^k\right) \propto \exp(\langle v_{w_{k+1}}, \frac{1}{k}\sum_{i=1}^k v_{w_i}\rangle).$$

This expression seems mysterious since it depends upon the *average* word vector for the previous k words. We show it can be theoretically justified. Assume a simplified version of our model, where a small window of k words is generated as follows: sample $c \sim \mathcal{C}$, where $\mathcal{C}$ is a uniformly random unit vector, then sample $(w_1, w_2, \ldots, w_k) \sim \exp(\langle \sum_{i=1}^k v_{w_i}, c\rangle)/Z_c$. Furthermore, assume $Z_c = Z$ for any c.

Lemma 3.1. *In the simplified version of our model, the Maximum-a-Posteriori (MAP) estimate of c given $(w_1, w_2, \ldots, w_k)$ is $\frac{\sum_{i=1}^k v_{w_i}}{\|\sum_{i=1}^k v_{w_i}\|_2}$.*

Proof. The c maximizing $p\left(c|w_1, w_2, \ldots, w_k\right)$ is the maximizer of $p(c)p\left(w_1, w_2, \ldots, w_k|c\right)$. Since $p(c) = p(c')$ for any c, c', and we have $p\left(w_1, w_2, \ldots, w_k|c\right) = \exp(\langle \sum_i v_{w_i}, c\rangle)/Z$, the maximizer is clearly $c = \frac{\sum_{i=1}^k v_{w_i}}{\|\sum_{i=1}^k v_{w_i}\|_2}$. □

Thus using the MAP estimate of c_t gives essentially the same expression as CBOW apart from the rescaling, which is often omitted due to computational efficiency in empirical works.

4 Explaining RELATIONS=LINES

As mentioned, word analogies like "a:b::c:??" can be solved via a linear algebraic expression:

$$\operatorname*{argmin}_d \|v_a - v_b - v_c + v_d\|_2^2, \tag{4.1}$$

where vectors have been normalized such that $\|v_d\|_2 = 1$. This suggests that the semantic relationships being tested in the analogy are characterized by a straight line,[5] referred to earlier as RELATIONS=LINES.

Using our model we will show the following for low-dimensional embeddings: for each such relation R there is a direction μ_R in space such that for any word pair a, b satisfying the relation, $v_a - v_b$ is like μ_R plus some noise vector. This happens for relations satisfying a certain condition described below. Empirical results supporting this theory appear in Section 5, where this linear structure is further leveraged to slightly improve analogy solving.

A side product of our argument will be a mathematical explanation of the empirically well-established superiority of low-dimensional word embeddings over high-dimensional ones in this setting (Levy and Goldberg, 2014a). As mentioned earlier, the usual explanation that smaller models generalize better is fallacious.

We first sketch what was missing in prior attempts to prove versions of RELATIONS=LINES from first principles. The basic issue is approximation error: the difference between the best solution and the 2nd best solution to (4.1) is typically small, whereas the approximation error in the objective in the low-dimensional solutions is larger. For instance, if one uses our **PMI** objective, then the weighted average of the termwise error in (2.5) is 17%, and the expression in (4.1) above contains six inner products. Thus in principle the approximation error could lead to a failure of the method and the emergence of linear relationship, but it does not.

Prior explanations. Pennington et al. (2014) try to propose a model where such linear relationships should occur *by design*. They posit that *queen* is a solution to the analogy "*man*:*woman*::*king*:??" because

$$\frac{p(\chi \mid king)}{p(\chi \mid queen)} \approx \frac{p(\chi \mid man)}{p(\chi \mid woman)}, \tag{4.2}$$

where $p(\chi \mid king)$ denotes the conditional probability of seeing word χ in a small window of text around $king$. Relationship (4.2) is intuitive since both sides will be ≈ 1 for gender-neutral χ like "*walks*" or "*food*", will be > 1 when χ is like "*he, Henry*" and will be < 1 when χ is like "*dress, she, Elizabeth*." This was also observed by Levy and Goldberg (2014a). Given (4.2), they then posit that the correct model describing word embeddings in terms of word occurrences must be a *homomorphism* from $(\Re^d, +)$ to $(\Re^+, \times)$, so vector differences map to ratios of probabilities. This leads to the expression

$$p_{w,w'} = \langle v_w, v_{w'} \rangle + b_w + b_{w'},$$

and their method is a (weighted) least squares fit for this expression. One shortcoming of this argument is that the homomorphism assumption *assumes* the linear relationships instead of explaining them from a more basic principle. More importantly, the empirical fit to the homomorphism has nontrivial approximation error, high enough that it does not imply the desired strong linear relationships.

Levy and Goldberg (2014b) show that empirically, skip-gram vectors satisfy

$$\langle v_w, v_{w'} \rangle \approx \text{PMI}(w, w') \tag{4.3}$$

up to some shift. They also give an argument suggesting this relationship must be present if the solution is allowed to be very high-dimensional. Unfortunately, that argument does not extend to low-dimensional embeddings. Even if it did, the issue of termwise approximation error remains.

Our explanation. The current paper has introduced a generative model to theoretically explain the emergence of relationship (4.3). However, as noted after Theorem 2.2, the issue of high approximation error does not go away either in theory or in the empirical fit. We now show that the isotropy of word vectors (assumed in the theoretical model and verified empirically) implies that even a weak version of (4.3) is enough to imply the emergence of the observed linear relationships in low-dimensional embeddings.

[5] Note that this interpretation has been disputed; e.g., it is argued in Levy and Goldberg (2014a) that (4.1) can be understood using only the classical connection between inner product and word similarity, using which the objective (4.1) is slightly improved to a different objective called 3COSMUL. However, this "explanation" is still dogged by the issue of large termwise error pinpointed here, since inner product is only a rough approximation to word similarity. Furthermore, the experiments in Section 5 clearly support the RELATIONS=LINES interpretation.

This argument will assume the analogy in question involves a relation that obeys Pennington et al.'s suggestion in (4.2). Namely, for such a relation R there exists function $\nu_R(\cdot)$ depending only upon R such that for any a, b satisfying R there is a *noise function* $\xi_{a,b,R}(\cdot)$ for which:

$$\frac{p(\chi \mid a)}{p(\chi \mid b)} = \nu_R(\chi) \cdot \xi_{a,b,R}(\chi) \tag{4.4}$$

For different words χ there is huge variation in (4.4), so the multiplicative noise may be large.

Our goal is to show that the low-dimensional word embeddings have the property that there is a vector μ_R such that for every pair of words a, b in that relation, $v_a - v_b = \mu_R +$ noise vector, where the noise vector is small.

Taking logarithms of (4.4) results in:

$$\log\left(\frac{p(\chi \mid a)}{p(\chi \mid b)}\right) = \log(\nu_R(\chi)) + \zeta_{a,b,R}(\chi) \tag{4.5}$$

Theorem 2.2 implies that the left-hand side simplifies to $\log\left(\frac{p(\chi|a)}{p(\chi|b)}\right) = \frac{1}{d}\langle v_\chi, v_a - v_b\rangle + \epsilon_{a,b}(\chi)$ where ϵ captures the small approximation errors induced by the inexactness of Theorem 2.2. This adds yet more noise! Denoting by V the $n \times d$ matrix whose rows are the v_χ vectors, we rewrite (4.5) as:

$$V(v_a - v_b) = d\log(\nu_R) + \zeta'_{a,b,R} \tag{4.6}$$

where $\log(\nu_R)$ in the element-wise log of vector ν_R and $\zeta'_{a,b,R} = d(\zeta_{a,b,R} - \epsilon_{a,b,R})$ is the noise.

In essence, (4.6) shows that $v_a - v_b$ is a solution to a linear regression in d variables and m constraints, with $\zeta'_{a,b,R}$ being the "noise." The *design matrix* in the regression is V, the matrix of all word vectors, which in our model (as well as empirically) satisfies an isotropy condition. This makes it random-like, and thus solving the regression by left-multiplying by $V^\dagger$, the pseudo-inverse of V, ought to "denoise" effectively. We now show that it does.

Our model assumed the set of all word vectors satisfies bulk properties similar to a set of Gaussian vectors. The next theorem will only need the following weaker properties. (1) The smallest nonzero singular value of V is larger than some constant c_1 times the quadratic mean of the singular values, namely, $\|V\|_F/\sqrt{d}$. Empirically we find $c_1 \approx 1/3$ holds; see Section 5. (2) The left singular vectors behave like random vectors with respect to $\zeta'_{a,b,R}$, namely, have inner product at most $c_2\|\zeta'_{a,b,R}\|/\sqrt{n}$ with $\zeta'_{a,b,R}$, for some constant c_2. (3) The max norm of a row in V is $O(\sqrt{d})$. The proof is included in the appendix.

Theorem 4.1 (Noise reduction)**.** *Under the conditions of the previous paragraph, the noise in the dimension-reduced semantic vector space satisfies*

$$\|\bar{\zeta}_{a,b,R}\|_2 \lesssim \|\zeta'_{a,b,R}\|_2 \frac{\sqrt{d}}{n}.$$

As a corollary, the relative error in the dimension-reduced space is a factor of $\sqrt{d/n}$ smaller.

5 Experimental verification

In this section, we provide experiments empirically supporting our generative model.

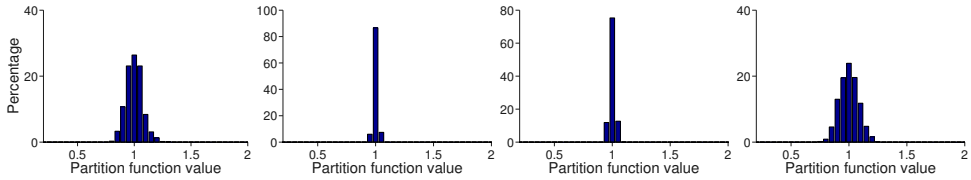
Corpus. All word embedding vectors are trained on the English Wikipedia (March 2015 dump). It is pre-processed by standard approach (removing non-textual elements, sentence splitting, and tokenization), leaving about 3 billion tokens. Words that appeared less than 1000 times in the corpus are ignored, resulting in a vocabulary of $68,430$. The co-occurrence is then computed using windows of 10 tokens to each side of the focus word.

Training method. Our embedding vectors are trained by optimizing the **SN** objective using AdaGrad (Duchi et al., 2011) with initial learning rate of 0.05 and 100 iterations. The **PMI** objective derived from (2.5) was also used. **SN** has average (weighted) term-wise error of 5%, and **PMI** has 17%. We observed that **SN** vectors typically fit the model better and have better performance, which can be explained by larger errors in PMI, as implied by Theorem 2.2. So, we only report the results for **SN**.

For comparison, GloVe and two variants of word2vec (skip-gram and CBOW) vectors are trained. GloVe's vectors are trained on the same co-occurrence as **SN** with the default parameter values.[6] word2vec vectors are trained using a window size of 10, with other parameters set to default values.[7]

[6] http://nlp.stanford.edu/projects/glove/

[7] https://code.google.com/p/word2vec/



(a) **SN** (b) GloVe (c) CBOW (d) skip-gram

Figure 1: The partition function Z_c. The figure shows the histogram of Z_c for 1000 random vectors c of appropriate norm, as defined in the text. The x-axis is normalized by the mean of the values. The values Z_c for different c concentrate around the mean, mostly in $[0.9, 1.1]$. This concentration phenomenon is predicted by our analysis.

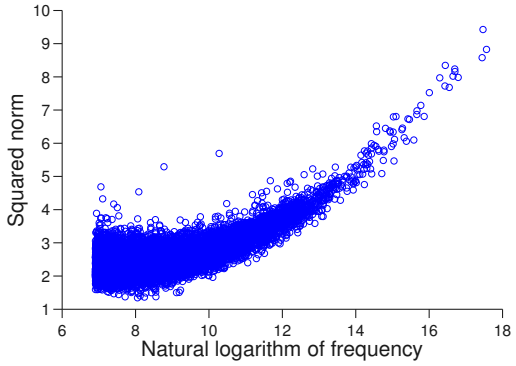


Figure 2: The linear relationship between the squared norms of our word vectors and the logarithms of the word frequencies. Each dot in the plot corresponds to a word, where x-axis is the natural logarithm of the word frequency, and y-axis is the squared norm of the word vector. The Pearson correlation coefficient between the two is 0.75, indicating a significant linear relationship, which strongly supports our mathematical prediction, that is, equation (2.4) of Theorem 2.2.

5.1 Model verification

Experiments were run to test our modeling assumptions. First, we tested two counter-intuitive properties: the concentration of the partition function Z_c for different discourse vectors c (see Theorem 2.1), and the random-like behavior of the matrix of word embeddings in terms of its singular values (see Theorem 4.1). For comparison we also tested these properties for word2vec and GloVe vectors, though they are trained by different objectives. Finally, we tested the linear relation between the squared norms of our word vectors and the logarithm of the word frequencies, as implied by Theorem 2.2.

Partition function. Our theory predicts the counter-intuitive concentration of the partition function $Z_c = \sum_{w'} \exp(c^\top v_{w'})$ for a random discourse vector c (see Lemma 2.1). This is verified empirically by picking a uniformly random direction, of norm $\|c\| = 4/\mu_w$, where μ_w is the average norm of the word vectors.[8] Figure 1(a) shows the histogram of Z_c for 1000 such randomly chosen c's for our vectors. The values are concentrated, mostly in the range $[0.9, 1.1]$ times the mean. Concentration is also observed for other types of vectors, especially for GloVe and CBOW.

Isotropy with respect to singular values. Our theoretical explanation of RELATIONS=LINES assumes that the matrix of word vectors behaves like a random matrix with respect to the properties of singular values. In our embeddings, the quadratic mean of the singular values is 34.3, while the minimum non-zero singular value of our word vectors is 11. Therefore, the ratio between them is a small constant, consistent with our model. The ratios for GloVe, CBOW, and skip-gram are 1.4, 10.1, and 3.1, respectively, which are also small constants.

Squared norms v.s. word frequencies. Figure 2 shows a scatter plot for the squared norms of our vectors and the logarithms of the word frequencies. A linear relationship is observed (Pearson correlation 0.75), thus supporting Theorem 2.2. The correlation is stronger for high frequency words, possibly because the corresponding terms have higher weights in the training objective.

This correlation is much weaker for other types

[8] Note that our model uses the inner products between the discourse vectors and word vectors, so it is invariant if the discourse vectors are scaled by s while the word vectors are scaled by $1/s$ for any $s > 0$. Therefore, one needs to choose the norm of c properly. We assume $\|c\|\mu_w = \sqrt{d}/\kappa \approx 4$ for a constant $\kappa = 5$ so that it gives a reasonable fit to the predicted dynamic range of word frequencies according to our theory; see model details in Section 2.

	Relations	**SN**	GloVe	CBOW	skip-gram
G	semantic	0.84	0.85	0.79	0.73
	syntactic	0.61	0.65	0.71	0.68
	total	0.71	0.73	0.74	0.70
M	adjective	0.50	0.56	0.58	0.58
	noun	0.69	0.70	0.56	0.58
	verb	0.48	0.53	0.64	0.56
	total	0.53	0.57	0.62	0.57

Table 1: The accuracy on two word analogy task testbeds: G (the GOOGLE testbed); M (the MSR testbed). Performance is close to the state of the art despite using a generative model with provable properties.

of word embeddings. This is possibly because they have more free parameters ("knobs to turn"), which imbue the embeddings with other properties. This can also cause the difference in the concentration of the partition function for the two methods.

5.2 Performance on analogy tasks

We compare the performance of our word vectors on analogy tasks, specifically the two testbeds GOOGLE and MSR (Mikolov et al., 2013a; Mikolov et al., 2013c). The former contains 7874 semantic questions such as "*man*:*woman*::*king*:??", and 10167 syntactic ones such as "*run*:*runs*::*walk*:??." The latter has 8000 syntactic questions for adjectives, nouns, and verbs.

To solve these tasks, we use linear algebraic queries.[9] That is, first normalize the vectors to unit norm and then solve "*a*:*b*::*c*:??" by

$$\operatorname*{argmin}_d \|v_a - v_b - v_c + v_d\|_2^2 . \qquad (5.1)$$

The algorithm succeeds if the best d happens to be correct.

The performance of different methods is presented in Table 1. Our vectors achieve performance comparable to the state of art on semantic analogies (similar accuracy as GloVe, better than word2vec). On syntactic tasks, they achieve accuracy 0.04 lower than GloVe and skip-gram, while CBOW typically outperforms the others.[10] The reason is probably that our model ignores local word order, whereas the other models capture it to some extent. For example, a word "*she*" can affect the context by a lot and determine if the next word is "*thinks*" rather than "*think*". Incorporating such linguistic features in the model is left for future work.

[9] One can instead use the 3COSMUL in (Levy and Goldberg, 2014a), which increases the accuracy by about 3%. But it is not linear while our focus here is the linear algebraic structure.

[10] It was earlier reported that skip-gram outperforms CBOW (Mikolov et al., 2013a; Pennington et al., 2014). This may be due to the different training data sets and hyperparameters used.

relation	cap-com	cap-wor	adj-adv	opp
1st	0.65 ± 0.07	0.61 ± 0.09	0.35 ± 0.17	0.42 ± 0.16
2nd	0.02 ± 0.28	0.00 ± 0.23	0.07 ± 0.24	0.01 ± 0.25

Table 2: The verification of relation directions on 2 semantic and 2 syntactic relations in the GOOGLE testbed. Relations include cap-com: capital-common-countries; cap-wor: capital-world; adj-adv: gram1-adjective-to-adverb; opp: gram2-opposite. For each relation, take $v_{ab} = v_a - v_b$ for pairs (a, b) in the relation, and then calculate the top singular vectors of the matrix formed by these v_{ab}'s. The row with label "1st"/"2nd" shows the cosine similarities of individual v_{ab} to the 1st/2nd singular vector (the mean and standard deviation).

5.3 Verifying RELATIONS=LINES

The theory in Section 4 predicts the existence of a direction for a relation, whereas earlier Levy and Goldberg (2014a) had questioned if this phenomenon is real. The experiment uses the analogy testbed, where each relation is tested using 20 or more analogies. For each relation, we take the set of vectors $v_{ab} = v_a - v_b$ where the word pair (a, b) satisfies the relation. Then calculate the top singular vectors of the matrix formed by these v_{ab}'s, and compute the cosine similarity (i.e., normalized inner product) of individual v_{ab} to the singular vectors. We observed that most $(v_a - v_b)$'s are correlated with the first singular vector, but have inner products around 0 with the second singular vector. Over all relations, the average projection on the first singular vector is 0.51 (semantic: 0.58; syntactic: 0.46), and the average on the second singular vector is 0.035. For example, Table 2 shows the mean similarities and standard deviations on the first and second singular vectors for 4 relations. Similar results are also obtained for word embedings by GloVe and word2vec. Therefore, the first singular vector can be taken as the direction associated with this relation, while the other components are like random noise, in line with our model.

	SN	GloVe	CBOW	skip-gram
w/o **RD**	0.71	0.73	0.74	0.70
RD(k = 20)	0.74	0.77	0.79	0.75
RD(k = 30)	0.79	0.80	0.82	0.80
RD(k = 40)	0.76	0.80	0.80	0.77

Table 3: The accuracy of the **RD** algorithm (i.e., the cheater method) on the GOOGLE testbed. The **RD** algorithm is described in the text. For comparison, the row "w/o **RD**" shows the accuracy of the old method without using **RD**.

	SN	GloVe	CBOW	skip-gram
w/o **RD-nn**	0.71	0.73	0.74	0.70
RD-nn (k = 10)	0.71	0.74	0.77	0.73
RD-nn (k = 20)	0.72	0.75	0.77	0.74
RD-nn (k = 30)	0.73	0.76	0.78	0.74

Table 4: The accuracy of the **RD-nn** algorithm on the GOOGLE testbed. The algorithm is described in the text. For comparison, the row "w/o **RD-nn**" shows the accuracy of the old method without using **RD-nn**.

Cheating solver for analogy testbeds. The above linear structure suggests a better (but cheating) way to solve the analogy task. This uses the fact that the same semantic relationship (e.g., masculine-feminine, singular-plural) is tested many times in the testbed. If a relation R is represented by a direction μ_R then the cheating algorithm can learn this direction (via rank 1 SVD) after seeing a few examples of the relationship. Then use the following method of solving "a:b::c:??": look for a word d such that $v_c - v_d$ has the largest projection on μ_R, the relation direction for (a, b). This can boost success rates by about 10%.

The testbed can try to combat such cheating by giving analogy questions in a random order. But the cheating algorithm can just *cluster* the presented analogies to learn which of them are in the same relation. Thus the final algorithm, named analogy solver with relation direction (**RD**), is: take all vectors $v_a - v_b$ for all the word pairs (a, b) presented among the analogy questions and do k-means clustering on them; for each (a, b), estimate the relation direction by taking the first singular vector of its cluster, and substitute that for $v_a - v_b$ in (5.1) when solving the analogy. Table 3 shows the performance on GOOGLE with different values of k; e.g. using our **SN** vectors and $k = 30$ leads to 0.79 accuracy. Thus future designers of analogy testbeds should remember not to test the same relationship too many times! This still leaves other ways to cheat, such as learning the directions for interesting semantic relations from other collections of analogies.

Non-cheating solver for analogy testbeds. Now we show that even if a relationship is tested only once in the testbed, there is a way to use the above structure. Given "a:b::c:??," the solver first finds the top 300 nearest neighbors of a and those of b, and then finds among these neighbors the top k pairs (a', b') so that the cosine similarities between $v_{a'} - v_{b'}$ and $v_a - v_b$ are largest. Finally, the solver uses these pairs to estimate the relation direction (via rank 1 SVD), and substitute this (corrected) estimate for $v_a - v_b$ in (5.1) when solving the analogy. This algorithm is named analogy solver with relation direction by nearest neighbors (**RD-nn**). Table 4 shows its performance, which consistently improves over the old method by about 3%.

6 Conclusions

A simple generative model has been introduced to explain the classical PMI based word embedding models, as well as recent variants involving energy-based models and matrix factorization. The model yields an optimization objective with essentially "no knobs to turn", yet the embeddings lead to good performance on analogy tasks, and fit other predictions of our generative model. A model with fewer knobs to turn should be seen as a better scientific explanation (*Occam's razor*), and certainly makes the embeddings more interpretable.

The spatial isotropy of word vectors is both an assumption in our model, and also a new empirical finding of our paper. We feel it may help with further development of language models. It is important for explaining the success of solving analogies via low dimensional vectors (RELATIONS=LINES). It also implies that semantic relationships among words manifest themselves as special directions among word embeddings (Section 4), which lead to a cheater algorithm for solving analogy testbeds.

Our model is tailored to capturing semantic similarity, more akin to a log-linear dynamic topic model. In particular, local word order is unim-

portant. Designing similar generative models (with provable and interpretable properties) with linguistic features is left for future work.

Acknowledgements

We thank the editors of TACL for granting a special relaxation of the page limit for our paper. We thank Yann LeCun, Christopher D. Manning, and Sham Kakade for helpful discussions at various stages of this work.

This work was supported in part by NSF grants CCF-1527371, DMS-1317308, Simons Investigator Award, Simons Collaboration Grant, and ONR-N00014-16-1-2329. Tengyu Ma was supported in addition by Simons Award in Theoretical Computer Science and IBM PhD Fellowship.

References

Jacob Andreas and Dan Klein. 2014. When and why are log-linear models self-normalizing? In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2015. A latent variable model approach to PMI-based word embeddings. Technical report, ArXiV. http://arxiv.org/abs/1502.03520.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. Linear algebraic structure of word senses, with applications to polysemy. Technical report, ArXiV. http://arxiv.org/abs/1502.03520.

David Belanger and Sham M. Kakade. 2015. A linear dynamical system model for text. In *Proceedings of the 32nd International Conference on Machine Learning*.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*.

Fischer Black and Myron Scholes. 1973. The pricing of options and corporate liabilities. *Journal of Political Economy*.

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*.

David M. Blei. 2012. Probabilistic topic models. *Communication of the Association for Computing Machinery*.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*.

Shay B. Cohen, Karl Stratos, Michael Collins, Dean P. Foster, and Lyle Ungar. 2012. Spectral learning of latent-variable PCFGs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*.

Ronan Collobert and Jason Weston. 2008a. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*.

Ronan Collobert and Jason Weston. 2008b. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*.

John Rupert Firth. 1957. *A synopsis of linguistic theory*.

Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. 2007. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*.

Tatsunori B. Hashimoto, David Alvarez-Melis, and Tommi S. Jaakkola. 2016. Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics*.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. 2012. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*.

Omer Levy and Yoav Goldberg. 2014a. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics*.

Yariv Maron, Michael Lamar, and Elie Bienenstock. 2010. Sphere embedding: An application to part-of-speech induction. In *Advances in Neural Information Processing Systems*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *Proceedings of the International Conference on Learning Representations*.

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*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*.

Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. 1998. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the 7th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing*.

Douglas L. T. Rohde, Laura M. Gonnerman, and David C. Plaut. 2006. An improved model of semantic similarity based on lexical co-occurence. *Communication of the Association for Computing Machinery*.

David E. Rumelhart, Geoffrey E. Hinton, and James L. McClelland, editors. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1988. Learning representations by back-propagating errors. *Cognitive modeling*.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*.

A Proof sketches

Here we provide the proof sketches, while the complete proof can be found in the full version (Arora et al., 2015).

Proof sketch of Theorem 2.2 Let w and w' be two arbitrary words. Let c and c' denote two consecutive context vectors, where $c \sim \mathcal{C}$ and $c'|c$ is defined by the Markov kernel $p(c' \mid c)$.

We start by using the law of total expectation, integrating out the hidden variables c and c':

$$\begin{aligned} p(w,w') &= \mathop{\mathbb{E}}_{c,c'} \left[\Pr[w,w'|c,c']\right] \\ &= \mathop{\mathbb{E}}_{c,c'} [p(w|c)p(w'|c')] \\ &= \mathop{\mathbb{E}}_{c,c'} \left[\frac{\exp(\langle v_w, c\rangle)}{Z_c} \frac{\exp(\langle v_{w'}, c'\rangle)}{Z_{c'}}\right] \end{aligned} \tag{A.1}$$

An expectation like (A.1) would normally be difficult to analyze because of the partition functions. However, we can assume the inequality (2.2), that is, the partition function typically does not vary much for most of context vectors c. Let F be the event that both c and c' are within $(1 \pm \epsilon_z)Z$. Then by (2.2) and the union bound, event F happens with probability at least $1 - 2\exp(-\Omega(\log^2 n))$. We will split the right-hand side (RHS) of (A.1) into the parts according to whether F happens or not.

$$\begin{aligned} \text{RHS of (A.1)} = &\underbrace{\mathop{\mathbb{E}}_{c,c'} \left[\frac{\exp(\langle v_w, c\rangle)}{Z_c} \frac{\exp(\langle v_{w'}, c'\rangle)}{Z_{c'}} \mathbf{1}_F\right]}_{T_1} \\ &+ \underbrace{\mathop{\mathbb{E}}_{c,c'} \left[\frac{\exp(\langle v_w, c\rangle)}{Z_c} \frac{\exp(\langle v_{w'}, c'\rangle)}{Z_{c'}} \mathbf{1}_{\bar{F}}\right]}_{T_2} \end{aligned} \tag{A.2}$$

where $\bar{F}$ denotes the complement of event F and $\mathbf{1}_F$ and $\mathbf{1}_{\bar{F}}$ denote indicator functions for F and $\bar{F}$, respectively. When F happens, we can replace Z_c by Z with a $1 \pm \epsilon_z$ factor loss: The first term of the RHS of (A.2) equals to

$$T_1 = \frac{1 \pm O(\epsilon_z)}{Z^2} \mathop{\mathbb{E}}_{c,c'} \left[\exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) \mathbf{1}_F\right] \tag{A.3}$$

On the other hand, we can use $\mathbb{E}[\mathbf{1}_{\bar{F}}] = \Pr[\bar{F}] \leq \exp(-\Omega(\log^2 n))$ to show that the second term of RHS of (A.2) is negligible,

$$|T_2| = \exp(-\Omega(\log^{1.8} n)) \,. \tag{A.4}$$

This claim must be handled somewhat carefully since the RHS does not depend on d at all. Briefly, the reason this holds is as follows: in the regime when d is small ($\sqrt{d} = o(\log^2 n)$), any word vector v_w and discourse c satisfies that $\exp(\langle v_w, c\rangle) \le \exp(\|v_w\|) = \exp(O(\sqrt{d}))$, and since $\mathbb{E}[\mathbf{1}_{\bar{F}}] = \exp(-\Omega(\log^2 n))$, the claim follows directly; In the regime when d is large ($\sqrt{d} = \Omega(\log^2 n)$), we can use concentration inequalities to show that except with a small probability $\exp(-\Omega(d)) = \exp(-\Omega(\log^2 n))$, a uniform sample from the sphere behaves equivalently to sampling all of the coordinates from a standard Gaussian distribution with mean 0 and variance $\frac{1}{d}$, in which case the claim is not too difficult to show using Gaussian tail bounds.

Therefore it suffices to only consider (A.3). Our model assumptions state that c and c' cannot be too different. We leverage that by rewriting (A.3) a little, and get that it equals

$$T_1 = \frac{1 \pm O(\epsilon_z)}{Z^2} \mathop{\mathbb{E}}_{c} \left[\exp(\langle v_w, c\rangle) \mathop{\mathbb{E}}_{c'|c} \left[\exp(\langle v_{w'}, c'\rangle) \right] \right]$$
$$= \frac{1 \pm O(\epsilon_z)}{Z^2} \mathop{\mathbb{E}}_{c} \left[\exp(\langle v_w, c\rangle) A(c) \right] \tag{A.5}$$

where $A(c) := \mathop{\mathbb{E}}_{c'|c} \left[\exp(\langle v_{w'}, c'\rangle) \right]$. We claim that $A(c) = (1 \pm O(\epsilon_2)) \exp(\langle v_{w'}, c\rangle)$. Doing some algebraic manipulations,

$$A(c) = \exp(\langle v_{w'}, c\rangle) \mathop{\mathbb{E}}_{c'|c} \left[\exp(\langle v_{w'}, c' - c\rangle) \right].$$

Furthermore, by our model assumptions, $\|c - c'\| \le \epsilon_2/\sqrt{d}$. So

$$\langle v_w, c - c'\rangle \le \|v_w\| \|c - c'\| = O(\epsilon_2)$$

and thus $A(c) = (1 \pm O(\epsilon_2)) \exp(\langle v_{w'}, c\rangle)$. Plugging the simplification of $A(c)$ to (A.5),

$$T_1 = \frac{1 \pm O(\epsilon_z)}{Z^2} \mathbb{E}[\exp(\langle v_w + v_{w'}, c\rangle)]. \tag{A.6}$$

Since c has uniform distribution over the sphere, the random variable $\langle v_w + v_{w'}, c\rangle$ has distribution pretty similar to Gaussian distribution $\mathcal{N}(0, \|v_w + v_{w'}\|^2/d)$, especially when d is relatively large. Observe that $\mathbb{E}[\exp(X)]$ has a closed form for Gaussian random variable $X \sim \mathcal{N}(0, \sigma^2)$,

$$\mathbb{E}[\exp(X)] = \int_x \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2}) \exp(x) dx$$
$$= \exp(\sigma^2/2). \tag{A.7}$$

Bounding the difference between $\langle v_w + v_{w'}, c\rangle$ from Gaussian random variable, we can show that for $\epsilon = \widetilde{O}(1/d)$,

$$\mathbb{E}[\exp(\langle v_w + v_{w'}, c\rangle)] = (1 \pm \epsilon) \exp\left(\frac{\|v_w + v_{w'}\|^2}{2d} \right). \tag{A.8}$$

Therefore, the series of simplification/approximation above (concretely, combining equations (A.1), (A.2), (A.4), (A.6), and (A.8)) lead to the desired bound on $\log p(w, w')$ for the case when the window size $q = 2$. The bound on $\log p(w)$ can be shown similarly.

Proof sketch of Lemma 2.1 Note that for fixed c, when word vectors have Gaussian priors assumed as in our model, $Z_c = \sum_w \exp(\langle v_w, c\rangle)$ is a sum of independent random variables.

We first claim that using proper concentration of measure tools, it can be shown that the variance of Z_c are relatively small compared to its mean $\mathbb{E}_{v_w}[Z_c]$, and thus Z_c concentrates around its mean. Note this is quite non-trivial: the random variable $\exp(\langle v_w, c\rangle)$ is neither bounded nor subgaussian/sub-exponential, since the tail is approximately inverse poly-logarithmic instead of inverse exponential. In fact, the same concentration phenomenon does not happen for w. The occurrence probability of word w is not necessarily concentrated because the ℓ_2 norm of v_w can vary a lot in our model, which allows the frequency of the words to have a large dynamic range.

So now it suffices to show that $\mathbb{E}_{v_w}[Z_c]$ for different c are close to each other. Using the fact that the word vector directions have a Gaussian distribution, $\mathbb{E}_{v_w}[Z_c]$ turns out to only depend on the norm of c (which is equal to 1). More precisely,

$$\mathop{\mathbb{E}}_{v_w}[Z_c] = f(\|c\|_2^2) = f(1) \tag{A.9}$$

where f is defined as $f(\alpha) = n\,\mathbb{E}_s[\exp(s^2\alpha/2)]$ and s has the same distribution as the norms of the word

vectors. We sketch the proof of this. In our model, $v_w = s_w \cdot \hat{v}_w$, where $\hat{v}_w$ is a Gaussian vector with identity covariance I. Then

$$\begin{aligned}\mathop{\mathbb{E}}_{v_w}[Z_c] &= n \mathop{\mathbb{E}}_{v_w}[\exp(\langle v_w, c\rangle)] \\ &= n \mathop{\mathbb{E}}_{s_w}\left[\mathop{\mathbb{E}}_{v_w|s_w}[\exp(\langle v_w, c\rangle) \mid s_w]\right]\end{aligned}$$

where the second line is just an application of the law of total expectation, if we pick the norm of the (random) vector v_w first, followed by its direction. Conditioned on s_w, $\langle v_w, c\rangle$ is a Gaussian random variable with variance $\|c\|_2^2 s_w^2$, and therefore using similar calculation as in (A.7), we have

$$\mathop{\mathbb{E}}_{v_w|s_w}[\exp(\langle v_w, c\rangle) \mid s_w] = \exp(s^2\|c\|_2^2/2)\,.$$

Hence, $\mathbb{E}_{v_w}[Z_c] = n\,\mathbb{E}_s[\exp(s^2\|c\|_2^2/2)]$ as needed.

Proof of Theorem 4.1 The proof uses the standard analysis of linear regression. Let $V = P\Sigma Q^T$ be the SVD of V and let $\sigma_1, \ldots, \sigma_d$ be the left singular values of V (the diagonal entries of Σ). For notational ease we omit the subscripts in $\bar{\zeta}$ and ζ' since they are not relevant for this proof. Since $V^\dagger = Q\Sigma^{-1}P^T$ and thus $\bar{\zeta} = V^\dagger\zeta' = Q\Sigma^{-1}P^T\zeta'$, we have

$$\|\bar{\zeta}\|_2 \le \sigma_d^{-1}\|P^T\zeta'\|_2. \tag{A.10}$$

We claim

$$\sigma_d^{-1} \le \sqrt{\frac{1}{c_1 n}}. \tag{A.11}$$

Indeed, $\sum_{i=1}^d \sigma_i^2 = O(nd)$, since the average squared norm of a word vector is d. The claim then follows from the first assumption. Furthermore, by the second assumption, $\|P^T\zeta'\|_\infty \le \frac{c_2}{\sqrt{n}}\|\zeta'\|_2$, so

$$\|P^T\zeta'\|_2^2 \le \frac{c_2^2 d}{n}\|\zeta'\|_2^2. \tag{A.12}$$

Plugging (A.11) and (A.12) into (A.10), we get

$$\|\bar{\zeta}\|_2 \le \sqrt{\frac{1}{c_1 n}}\sqrt{\frac{c_2^2 d}{n}\|\zeta'\|_2^2} = \frac{c_2\sqrt{d}}{\sqrt{c_1}n}\|\zeta'\|_2$$

as desired. The last statement follows because the norm of the signal, which is $d\log(\nu_R)$ originally and is $V^\dagger d\log(\nu_R) = v_a - v_b$ after dimension reduction, also gets reduced by a factor of $\sqrt{n}$.