Random Walks on Context Spaces: Towards an Explanation of the Mysteries of Semantic Word Embeddings

Sanjeev Arora * Yuanzhi Li † Yingyu Liang ‡ Tengyu Ma § Andrej Risteski ¶

February 13, 2015

Abstract

The papers of Mikolov et al. 2013 as well as subsequent works have led to dramatic progress in solving word analogy tasks using semantic word embeddings. This leverages *linear structure* that is often found in the word embeddings, which is surprising since the training method is usually nonlinear. There were attempts —notably by Levy and Goldberg and Pennington et al.— to explain how this linear structure arises. The current paper points out the gaps in these explanations and provides a more complete explanation using a *loglinear generative* model for the corpus that directly models the latent semantic structure in words. The novel methodological twist is that instead of trying to fit the best model parameters to the data, a rigorous mathematical analysis is performed using the model priors to arrive at a simple closed form expression that approximately relates co-occurrence statistics and word embeddings. This expression closely corresponds to —and a bit simpler than— the existing training methods, and leads to good solutions to analogy tasks. Empirical support is provided also for the validity of the modeling assumptions.

This methodology of letting some mathematical analysis substitute for some of the computational difficulty may be useful in other settings with generative models.

1 Introduction

Embeddings of words as vectors in a relatively low-dimensional space go back several decades in linguistics (Deerwester et al., 1990; Hinton, 1984). Building such representations follows the well-known philosophy that the meaning of a word is defined by "the company it keeps," namely, co-occurrence statistics (Firth, 1957) (see (Rohde et al., 2006) for a survey). Past methods to obtain word embeddings include matrix factorization methods (e.g., (Deerwester et al., 1990)) neural networks (e.g., (Rumelhart et al., 1988; Bengio et al., 2006; Collobert and Weston, 2008)) and energy-based models.

A surprising discovery of Mikolov et al. (2013a;b) was that word embeddings created by a recursive neural net (RNN) as well as by a related energy-based model called **word2vec** exhibit additional *linear* structure, which allows easy solutions to analogy questions of the form "man:woman::king:??." Specifically, queen happens to be the word whose vector v_{queen} is most similar to the vector $v_{king} - v_{man} + v_{woman}$. (Note that the two vectors may only make an angle of say 45 degrees, but that is still a significant overlap in 300-dimensional space.)

A flurry of subsequent work exhibited similar linear structure in embeddings obtained from other methods: noise-contrastive estimation (Mnih and Kavukcuoglu, 2013), a specific weighted least squares model that trains on the *logarithm* of word-word co-occurrence counts (Pennington et al., 2014), and large-dimensional embeddings that explicitly encode co-occurrence statistics (Levy and Goldberg, 2014a).

*Princeton University, Computer Science Department. Email: arora@cs.princeton.edu.
†Princeton University, Computer Science Department. Email: yuanzhil@cs.princeton.edu
‡Princeton University, Computer Science Department. Email: yingyul@cs.princeton.edu
§Princeton University, Computer Science Department. Email: tengyu@cs.princeton.edu
¶Princeton University, Computer Science Department. Email: risteski@cs.princeton.edu

This phenomenon is mysterious at several levels. First, linear relationships emerge organically in radically different embedding methods, including highly nonlinear ones. Second, different methods yield fairly similar success rates on analogy tasks. Together, these suggest that the linear structure must be somehow *inherent in the data itself*. Levy and Goldberg (2014a) and Pennington et al. (2014) provide some intuitive justification but these explanations are incomplete, as we will see. We give a new explanation that is also supported using a new loglinear generative model for text data. While the model is interesting in itself (since all past work was in the discriminative setting), the more important contribution is our analysis technique. Instead of following the usual method of fitting the best model parameters to the data by bayesian methods, we do first a rigorous mathematical analysis using the model priors to arrive at a simple closed form expression that approximately relates co-occurrence statistics and word embeddings. This expression exactly corresponds to the training methods suggested –with heuristic justifications—in (Levy and Goldberg, 2014a) and (Pennington et al., 2014).

This generative model and its rigorous analysis may be useful in other domains where loglinear models are used. As a side product our generative model also suggests perhaps the simplest method thus far for finding word embeddings —related to but slightly simpler than the GLOVE model (Pennington et al., 2014)—that also solves analogy tasks pretty well (as reported in Section 5 and 6). The notable feature here is that this training *provably* finds the near-optimum fit to the generative model.

2 Semantic embeddings: the mysteries and the explanation

A sequence of papers by Mikolov et al. (2011; 2013a;b) culminated in the skip-gram with negative-sampling (SGNS) method, which we describe in the simplified formulation of (Goldberg and Levy, 2014). It utilizes co-occurence statistics of *words* and *contexts*. A simple example of *context* for occurrence of a word w is a pair (w', k) where word w' occurs with offset $k \in \{-2, -1, 1, 2\}$ from w. (Other types of contexts can be considered.) Their distribution is given a discriminative model:

$$\Pr(D = 1|w, \chi) = \frac{1}{1 + \exp\left(-\langle v_w, v_c \rangle\right)} \tag{2.1}$$

where v_w is the vector for word w and v_χ for context χ and the event $D = 1$ means the pair (w, χ) is observed in the corpus and $D = 0$ means it is not.

SGNS tries to maximize $\Pr(D = 1|w, \chi)$ for observed (w, χ) pairs while maximizing $\Pr(D = 0|w, \chi)$ for randomly sampled "negative" examples, under the assumption that randomly selecting a context for a given word is likely to result in an unobserved (w, χ) pair. The objective is the log likelihood.

The GloVe method (Pennington et al., 2014) does a more direct fit. If $X_{w,w'}$ denotes the co-occurrence count for a pair of words w and w', GloVe finds for each word w two low dimensional vectors $v_w, \tilde{v}_w$ and scalar $b_w, \tilde{b}_w$ so as to minimize:

$$\sum_{w,w'} f(X_{w,w'})(\langle v_w, \tilde{v}_{w'} \rangle - b_w - \tilde{b}_{w'} - \log X_{w,w'})^2 \tag{2.2}$$

where $f(x) = \min\{(x/x_{\max})^{0.75}, 1\}$ with $x_{\max} = 100$.

Once word vectors have been produced, the query to find solution d for the analogy task "a:b::c:??" is the following, where vectors have been normalized so v_d is a unit vector:

$$\begin{aligned} d &= \operatorname*{argmax}_d \langle v_d, v_c + v_b - v_a \rangle && (2.3) \\ &= \operatorname*{argmin}_d \|v_a - v_b - v_c + v_d\|_2^2 && (2.4) \end{aligned}$$

where the equality follows from $\|v_a - v_b - v_c + v_d\|_2^2 = \|v_a - v_b - v_c\|_2^2 + \|v_d\|_2^2 + 2\langle v_a - v_b - v_c, v_d \rangle$ and $\|v_d\|_2^2 = 1$.

We note that in the experiments essentially the same performance is achieved by (2.4) with unnormalized vectors, which is more convenient for our analysis. Therefore, we will focus on this query in the following discussions.

Intuitive justification: Both Pennington et al. (2014) and Levy and Goldberg (2014a) describe the statistics intuition why the answer to the analogy "man:woman::king:??" must be *queen*. The reason is that most contexts χ satisfy

$$\frac{p(\chi \mid king)}{p(\chi \mid queen)} \approx \frac{p(\chi \mid man)}{p(\chi \mid woman)}.$$

where $p(w, \chi)$ is the co-occurrence frequency of the pair (w, χ). Indeed, both ratios will be around 1 for most contexts, e.g., *walks* or *food*, but will deviate from 1 when χ involves, say, *dress, he*, *she*, *Elizabeth*, *Henry*, etc. Therefore, a reasonable strategy to solve the analogy is to find a word w that minimizes

$$\sum_{\chi} \left(\log \left(\frac{p(\chi \mid king)}{p(\chi \mid queen)} \right) - \log \left(\frac{p(\chi \mid man)}{p(\chi \mid woman)} \right) \right)^2 \tag{2.5}$$

Levy and Goldberg (2014a) therefore proposed a simple but high-dimensional embedding (introduced by Church and Hanks (1990)) that explicitly encodes correlation statistics between words and contexts. The vector for word w is indexed by all possible contexts, and the entry in coordinate χ is $PMI(w, \chi)$, which is defined as $\log \frac{p(w,\chi)}{p(w)p(\chi)}$. With this word embedding, the query expression (2.4) is easily verified to be *equivalent* to (2.5). Thus explicit word embeddings should indeed solve analogies via linear algebraic queries, and they do empirically.

The attempted unification: The above intuition only applies to a very high-dimensional embeddings that explicitly encode correlations. Subsequently Levy and Goldberg (2014b) suggested that even methods for producing low-dimensional embeddings must be capturing the essence of the above structure. Concretely, they suggested that current methods yield vector embeddings for context χ and word w satisfying

$$\langle v_w, v_\chi \rangle \approx PMI(w, \chi). \tag{2.6}$$

(Similar postulates for cooccurence data occured in (Globerson et al., 2007), and of course in GloVe; see (2.2).) They gave a heuristic argument that methods such as SGNS are implicitly doing this matrix factorization. But mathematically their argument is quite incomplete since they were unable to argue about the actual gradient and Hessian.

Even if Levy and Goldberg's intuitive argument about SGNS (Levy and Goldberg, 2014b) could be made rigorous (namely, that it does amount to low-rank matrix factorization) **Mystery 1** would still remain: why does this low-rank approximation help solve analogy tasks via the above linear algebraic queries? The inner product of Levy and Goldberg's explicit vectors for w, w' is $\sum_\chi PMI(w, \chi)PMI(w', \chi)$, which seems to have no connection to inner product of word vectors obtained by the matrix approximation in (2.6). Thus the matrix approximation does not seem to imply the hoped-for equivalence between the actual query (2.4) and (2.5). A further **Mystery 2** is that low-rank approximation to a matrix can in principle have high entry-wise error —our experiments show it is 5% or even higher per entry in (2.6)—and the query (2.4) involves 3 such inner products. A total error of 15% could in principle cripple the method but in practice it doesn't (Section 6 shows that the gap between the best and the second-best solution is not very large). We will try to explain this as well as the mysterious *shifts* used in previous training objectives—presumably discovered through trial and error—as in (2.2).

2.1 Our explanation of Mysteries 1 and 2

For ease of exposition we simplify the notion of *context* so that a context is just any word w' appearing in a small window around w. (Using more complicated notions of context provides only a small improvement on analogy tasks.) This simplifies the PMI matrix proposed by Levy and Goldberg (2014b) and makes it symmetric: each entry is indexed by pairs of words (w, w') and contains $PMI(w, w')$. Let n denotes the total number of words. We postulate the following.

Property A. Similar to (2.6), the PMI matrix is close to a positive semidefinite matrix of fairly low rank, closer to $\log n$ than to n. This postulate yields natural word embeddings that are implicit in the cooccurence data itself: $PMI(w, w') \approx \langle v_w, v_{w'} \rangle$.

Property B: The word vectors implicit in property A are approximately *isotropic*, meaning they have a fairly uniform spatial distribution. We will use the standard mathematical formalization of spatial uniformity, namely, that $\mathbb{E}_w[v_w v_w^\top]$ is approximately like the identity matrix I, in that every one of its eigenvalues lies in $[1, 1+\delta]$ for some small $\delta > 0$.

Section 3 gives a plausible *generative model* for text corpora using log linear distributions, under which both properties hold. Theorem 1 shows that (up to a constant shift $\log Z$ and a some small error)

$$\begin{aligned} & \log p(w, w') \propto \|v_w + v'_w\|^2 - 2 \log Z \\ \text{and} \qquad & \log p(w) \propto \|v_w\|^2 - \log Z \end{aligned} \tag{2.7}$$

so we conclude that up to some error $PMI(w, w') \propto \langle v_w, v_{w'} \rangle$. (This is the basis for our training method in Section 6.) The vectors v_w are isotropic by model assumptions, so Property B holds.

The explanation to **Mystery 1** uses Property B, specifically the following mathematical consequence of isotropy: denoting $\Sigma = \mathbb{E}_w[v_w v_w^T]$ we have for every vector v

$$\begin{aligned} \|v\|^2 &\approx v^T \Sigma v \qquad \text{(up to error } 1+\delta) \\ &= \mathbb{E}_w[\langle v, v_w \rangle^2]. \end{aligned}$$

Thus the query in (2.4) for solving analogy tasks turns into

$$\underset{d}{\operatorname{argmin}} \|v_a - v_b - v_c + v_d\|_2^2 \tag{2.8}$$

$$\approx \underset{d}{\operatorname{argmin}} \mathbb{E}_w \left(\langle v_a, v_w \rangle - \langle v_b, v_w \rangle - \langle v_c, v_w \rangle + \langle v_d, v_w \rangle \right)^2 \tag{2.9}$$

The right hand side of the last expression is in turn close to (2.5) (where word w acts as context χ) since matrix approximation tries to ensure $\langle v_a, v_w \rangle \approx PMI(a, w)$. This explains **Mystery 1.** Of course, (2.8) only has to hold for the query vectors and not all vectors v, so weaker forms of isotropy could also suffice. (Empirically, we find the Isotropy as defined holds with fairly small δ see Section 6.)

The above explanation also helps explain **Mystery 2** since it shows that the query is effectively an average over many contexts w's, and this averaging helps lower error. Specifically, our training is using the *empirical* value of $PMI()$ whereas the model applies to some "ground truth "value in an infinitely large corpus. Thus each inner product on the right hand side of (2.9) has an inherent error due to sampling even if the matrix approximation error were perfect. A simple argument (assuming sampling errors in entries are pairwise independent and expectation 0) shows that the above averaging should reduce this error. Thus isotropy of word vectors accounts for the stability of the entire setup.

The above description also clarifies the possible sources of overtraining that we actually observe: (i) Theorem (2.7) only predicts an approximate low rank factorization of the PMI matrix so trying to fit too closely doesn't help and may hurt. (ii) If the word vectors are not low dimensional, they may not be isotropic and hence linear algebraic queries like (2.4) no longer have an interpretation as approximate queries like (2.9) to the word distribution. (iii) even if the model were exact, the empirical PMI matrix is only an approximation to the ground truth PMI matrix, and the smaller a probability, the noisier the empirical estimate. This means that instead of a low rank symmetric factorization (in other words, SVD) one needs a *weighted* analog similar to (2.2). (See Section 5.) Technically this is NP-hard, but in this setting gradient descent methods seem to do well. We further note that the Johnson-Lindenstrauss transform can also be used to project the explicit vectors to low dimension while approximately preserving the inner product. However, it has a quadratic dependence on the accuracy paparameter, so will not lead to vectors with as low a dimension as low rank matrix factorization approaches will. Furthermore, it does not take into account the weights.

3 Generative model

Markovian models with loglinear distributions are ubiquitous in language processing. Usually they express observables in terms of past observables, such as expressing the probability of the next word as a function of the previous two words. The intuition underlying semantic vector embeddings is that correlation among observables captures underlying semantics. Here we try to capture this intuition more directly using a generative model for text corpora that directly models *latent* semantic structure. Semantics is captured by a real-valued vector called *context*, denoted $c \in \Re^d$. The context vector c_t at time t determines the probability of generating various words at that time. The context vector undergoes continuous *drift* as the corpus gets generated, and furthermore, this drift is a *random walk* in $\Re^d$. The coordinates of the context vector represent *topics*. If the ith coordinate of the context corresponds to *gender*, its value represents the extent to which *gender* is being talked about at the moment. A positive value of this coordinate could correspond to maleness —leading to an increase in the probability of producing words like *he, king, man*— and negative value could correspond to femaleness —causing a probability increase for words like *she, queen, woman*.

To capture this word production behavior we use a log linear model: every word has a (time-invariant) vector representation that captures its correlations with topics. Thus *king* could have a positive value in the coordinate corresponding to gender and *queen* could have a negative value. If v_w is a vector for a word w and c_t is the current context vector, the probability of generating word w at time t is proportional to $\exp(\langle c_t, v_w \rangle)$. Since the context vector only drifts slowly with time, this distribution is fairly stable in a small window of text. Thus words w', w that both overlap well with c_t will tend to co-occur at time t. If $v_w, v_{w'}$ overlap well with many contexts then this suggests some overlap between $v_{w'}, v_w$ as well. Thus intuitively, cooccurence in the corpus captures some semantic similarity, though the precise relationship has to be worked out.

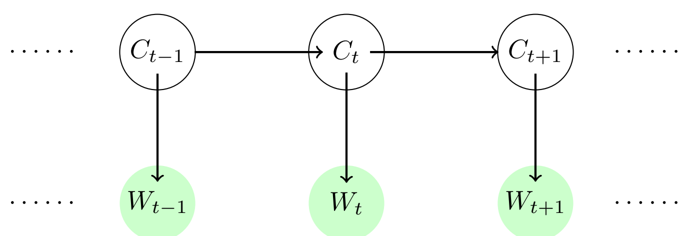


Figure 1: Illustration of the HMM.

Combining familiar elements like HMMs and loglinear distribution, the above model is not surprising *per se*. The *dynamic topic model* of (Blei and Lafferty, 2006) uses a similar notion of drift of topics, except the word production model is linear in the context vector instead of loglinear. Lafferty's linear chain CRF is a more general loglinear model.

In fact we suspect others must have considered this model and discarded it for the following reasons: (a) computational cost for such large corpora (b) the number of latent context vectors needs to be larger than the corpus size (our model even allows them to be infinite) so the training would not be meaningful.

The novel twist here is what we do with the model. Instead of fitting using usual bayesian optimization, we start with a specific prior for the markov chain (how the context vector drifts over time) and the model parameters (how the word vectors are distributed in context space). We then *analytically* compute a simpler closed form expression that approximately connects the model parameters to the observable statistics. This simpler relation makes it much easier to solve the model on very large datasets. (This is reminiscent of analyses of similar random walk models in finance such as Black-Scholes.) The analysis in some sense does the heavy lifting by doing away with the partition function. Recall that in a loglinear model $\Pr[w|c] \propto \exp(\langle v_w, c\rangle)$ the actual probability is given by

$$\Pr[w|c] = \frac{\exp(\langle v_w, c\rangle)}{\sum_w \exp(\langle v_w, c\rangle)} = \frac{1}{Z_c}\exp(\langle v_w, c\rangle); \tag{3.1}$$

where Z_c is the *partition function*, the usual source of much computational difficulty. Our analysis will show that Z_c does not vary much at all over time as the context shifts, which will simplify the rest of the calculation, allowing us to conclude the simple properties of Theorem 1.

Model details: If the number of words is n, the ambient dimension of the space is d, say $> \log^2 n$ and less than $\sqrt{n}$. The set of context vectors is continuous and the random walk has a stationary distribution on it that is a product distribution on coordinates. Concretely, say each coordinate is uniformly distributed in $[-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}]$. The walk can move in any direction with a probability that preserves the uniform distribution, but at each step the drift in the context vector[1] is much less than $1/\log n$ in ℓ_1 norm. This is still fast enough to let the walk explore the space quickly.

Word vectors need to have noticeably varying lengths, to fit the empirical fact that word probabilities satisfy a power law. We assume that word vectors are i.i.d generated by $v = s \cdot \hat{v}$ where $\hat{v}$ is from spherical gaussian distribution and s is a random scalar that has expectation and standard deviation less than $\sqrt{d}$ and is bounded by $\kappa\sqrt{d}$ for constant κ. Basically, $\exp(\kappa^2)$ turns out to be the dynamic range (i.e. max to min ratio) between the word probabilities, so κ is some constant, say, 10.

4 Co-occurrence probability

Notation For this section, w is a word, v_w its semantic vector, c is the context vector and T denotes the entire corpus, with $|T|$ as bigger than n and d.

As mentioned, we use a very simple notion of *context*, which is any word appearing in a small window around w. Thus we are interested in understanding say $p(w, w')$, the probability that two words w, w' occur as a consecutive pair of words in the corpus (the same analysis works for pairs that appear in a small window). Also let $p(w)$ be the probability that w occurs at a certain time. Since the random walk on context vectors is in its stationary distribution, these probabilities don't depend on time t.

The following theorem characterizes this probability in terms of the underlying word vectors, and directly leads to our training method in Section 6.

Theorem 1 *With high probability (over the choices of v_w's), we have that for any two different words w and w'*

$$\log p(w, w') = \frac{1}{2d}\|v_w + v_{w'}\|^2 - 2\log Z \pm o(1) \tag{4.1}$$

for some fixed constant Z. Moreover,

$$\log(p(w)) = \frac{1}{2d}\|v_w\|_2^2 - \log Z \pm o(1). \tag{4.2}$$

Note that since the word vectors have ℓ_2 norm roughly $\sqrt{d}$, for two typical word vectors $v_w, v_{w'}$, $\|v_w + v_{w'}\|^2$ is of the order of $\Theta(d)$. Therefore the noise level $o(1)$ is dominated by the leading term $\frac{1}{2d}\|v_w + v_{w'}\|^2$, which is what we are going to fit.

Let c be the hidden context that determines the probability of word w, and c' be the next one that determines w'. We use $p(c'|c)$ to denote the Markov kernel (transition matrix) of the markov chain. Let $\mathcal{C}$ be the stationary distribution of context vector c.

Towards deriving the expression (4.1) for $p(w, w')$, we marginalize over the contexts c, c' and then use the independence of w, w' conditioned on c, c',

$$\begin{aligned} p(w, w') &= \int_{c,c'} p(w|c)p(w'|c')p(c, c')dcdc' \\ &= \int_{c,c'} \frac{\exp(\langle v_w, c\rangle)}{Z_c} \frac{\exp(\langle v_{w'}, c'\rangle)}{Z_{c'}} p(c)p(c'|c)dcdc' \end{aligned} \tag{4.3}$$

[1] The proof extends to any a symmetric product distribution over the coordinates satisfying, and $\mathbb{E}_{c\sim\mathcal{C}}\left[|c_i|^2\right] = \frac{1}{d}$, and $|c_i| \le \frac{2}{\sqrt{d}}$ a.s., the steps are such that for all c_t, $\mathbb{E}_{p(c_{t+1}|c_t)}[\exp(4\kappa|c_{t+1} - c_t|_1 \log n)] \le 1 + \epsilon_2$

The main difficulty here is dealing with the notorious partition function Z_c which is a sum of exponential functions

$$Z_c = \sum_w \exp(\langle v_w, c\rangle).$$

We circumvent this issue by showing that the partition functions Z_c's should be all $1 + o(1)$ close to each other for most of contexts c. (The rest of the proof could go through if the Z_c's were allowed to vary a bit more, say within a factor 2.)

Theorem 2 *There exists Z such that with high probability $(1 - 4\exp(-d^{0.2}))$ over the choice of v_w's and c,*

$$(1 - o(1))Z \le Z_c \le (1 + o(1))Z.$$

The proof of Theorem 2 uses the fact that word vectors are evenly spread out in space, being scaled multiples of vectors from a Normal distribution. Thus concentration of measure will be used to show that Z_c is close to its mean. Furthermore, the means of Z_c's are close to each other because they mainly depend on the norms of c's, which are also fairly concentrated around 1. However, this is actually non-trivial: the random variable $\exp(\langle v_w, c\rangle)$ is not subguassian nor bounded, since the scaling of w and c is such that $\langle v_w, c\rangle$ is $\Theta(1)$, and therefore $\exp(\langle v_w, c\rangle)$ is at the non-linear regime.

In fact, the same concentration phenomenon doesn't happen for w (and it had better not!). The occurrence probability of word w, which is also a weighted sum of exponential forms, is not necessarily concentrated because the ℓ_2 norm of v_w can vary a lot in our model.

Equipped with Theorem 2, we give in the rest of the section a non-rigorous sketch of the proof of equation (4.1) of Theorem 1. (More rigorous proof appears in supplementary material.) We start by getting rid of the partition function in the expression (4.3) using Theorem 2:

$$\begin{aligned}&\text{RHS of (4.3)}\\ &= \frac{(1 \pm o(1))^2}{Z^2} \int_{c,c'} \exp(\langle v_w + v_{w'}, c\rangle) p(c) p(c'|c) dc dc'\end{aligned}$$

For the sake of demonstration of ideas, we assume that the context drifts extremely slowly: with probability extremely close to 1, context c' is equal to c. In this special case, we can further simplify the expression above by basically assuming $c = c'$,

$$\begin{aligned} p(w, w') =& \text{ RHS of (4.3)}\\ =& \frac{(1 \pm o(1))^2}{Z^2} \int_c \exp(\langle v_w + v_{w'}, c\rangle) p(c) dc\\ =& \frac{(1 \pm o(1))^2}{Z^2} \mathop{\mathbb{E}}_{c\sim\mathcal{C}} [\exp(\langle v_w + v_{w'}, c\rangle)] \end{aligned} \tag{4.4}$$

Then using the assumption that the stationary distribution $\mathcal{C}$ is a product distribution under which c has norm expected to be 1, we can show that

$$\mathop{\mathbb{E}}_{c\sim\mathcal{C}} [\exp(\langle v_w + v_{w'}, c\rangle)] \approx \exp(\|v_w + v_{w'}\|^2/(2d)) \tag{4.5}$$

Hence, by connecting (4.4) and (4.5), we have the desired result. In an analogous way, we can prove that log occurrence probability of word w is proportional to $\|w\|^2$ up to an $o(1)$ shift.

5 Optimization objectives

In this section, we design objective functions for constructing word vectors according to our theory. Theorem 1 says that after proper scaling, $\log p(w, w')$ is roughly $\|v_w + v_{w'}\|_2^2 + C$ where $C = -2\log Z$ is an unknown constant. The ground truth $p(w, w')$ is approximated by the empirical co-occurrence count

$X_{w,w'}$ in the corpus T. This suggests minimizing the sum of the differences between $\log(X_{w,w'}/|T|)$ and $\|v_w + v_{w'}\|_2^2 + C$ over word vectors v_w's and C.

However, the sampling noise in $X_{w,w'}$ is significant (especially after taking logs), and therefore we need to weight w, w' with some $f_{w,w'}$ to compensate the noise in $X_{w,w'}$, where $f_{w,w'}$ should be increasing in $X_{w,w'}$ since for larger $p_{w,w'}$, $\log(X_{w,w'})$ actually have smaller variance, even though $X_{w,w'}$ have larger one. Though we omit the easy calculation, one can use maximum likelihood weight $f_{w,w'} = \frac{X_{w,w'}}{|T| - X_{w,w'}}$, by assuming that $X_{w,w'}$ is generated from the Binomial distribution (see supplementary). We arrive at the following optimization named **SN** (**S**quared **N**orm):

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

Note that GloVE also used a similar weighting (though not explained) where $f_{w,w'}$ is chosen to be linear or sublinear in $X_{w,w'}$. (The sublinear value can theoretically arise if the frequency distribution of the subset of words that actually arise in analogy tasks is not representative.) In experiment, we adopt the weights $f_{w,w'}$ used in GloVe (2.2).

We also introduce two similar objectives for comparison. Since $\|v_w + v_{w'}\|_2^2 = 2\langle v_w, v_{w'}\rangle + \|v_{w'}\|^2 + \|v_{w'}\|^2$, we can introduce bias terms b_w to capture $\|v_w\|^2 + C/2$, resulting in the objective named **BIAS**:

$$\min_{\{v_w,b_w\}} \sum_{w,w'} f_{w,w'} \left(\log X_{w,w'} - \langle v_w, v_{w'}\rangle - b_w - b_{w'}\right)^2 .$$

Note that although this looks similar to GloVe's objective, we uses only one set of word vectors to fit the data, without the context vectors used in GloVe. Finally, Theorem 1 implies the PMI $\log \frac{p(w,w')}{p(w')p(w')}$ is roughly $\langle v_w, v_{w'}\rangle + C$ for some constant C, leading to the following objective named **PMI**:

$$\min_{\{v_w\},C} \sum_{w,w'} f_{w,w'} \left(\log \frac{X_{w,w'}|T|}{X_w X_{w'}} - \langle v_w, v_{w'}\rangle - C\right)^2$$

where $X_w = \sum_{w'} X_{w,w'}$ and $\log \frac{X_{w,w'}|T|}{X_w X_{w'}}$ is the empirical PMI.

6 Experiment

We present a selected subset of experimental results here, while the complete details can be found in the appendix.

Training method We used the English Wikipedia data[2], and preprocessed it by standard approach, resulting in about 3 billion tokens. We ignored words that appeared less than 1000 times in the corpus and obtained a vocabulary of 68,430. The co-occurrence is then computed by using GloVe's code, on which all the methods are trained by AdaGrad (Duchi et al., 2011).

Evaluation method The constructed vectors are evaluated on two standard testbeds for word analogy tasks (GOOGLE[3], MSR[4]) and a more challenging testbed. GOOGLE and MSR questions are answered correctly only if the correct missing term is ranked top 1; in our testbed if the missing term is among the top 10. GOOGLE contains semantic questions such as "man:woman::king:?" and syntactic ones such as "run:runs::walk:?." MSR includes syntactic questions for adjectives, nouns and verbs. Our testbed includes semantic questions collected from English courses, such as "lettuce:vegetable::apples:?." This testbed will be

[2] `http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2`
[3] `code.google.com/p/word2vec/source/browse/trunk/questions-words.txt`
[4] `research.microsoft.com/en-us/projects/rnn/`

	Relations	GloVe	**SN**	**BIAS**	**PMI**
G	semantic	84.54	81.13	83.77	80.98
	syntactic	64.62	61.15	61.66	61.19
	total	73.32	69.87	71.31	69.82
M	adjective	54.01	50.00	51.81	49.24
	noun	73.10	69.70	70.50	68.60
	verb	59.43	47.70	48.73	48.33
	total	61.01	52.54	53.68	52.54
ours	total(@top10)	35.09	35.67	35.09	30.41

Table 1: Accuracy on the word analogy tasks. G: GOOGLE; M: MSR.

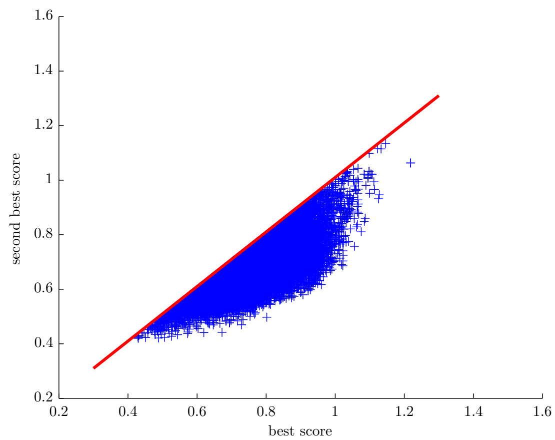


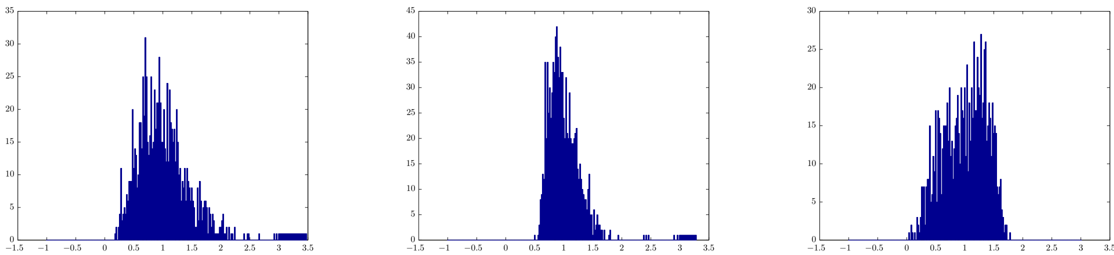
Figure 2: The scatter plot showing the small gap between the best score and the second best for questions in GOOGLE. The red line is $x = y$.

released with the public version of the paper. Our vocabulary covers about 92% of the GOOGLE questions, 63% of MSR, and 95% of our testbed. To solve these tasks, we use linear algebraic query (2.3), *i.e.*, rank d according to $\langle v_d, v_c + v_b - v_a \rangle$. We compare to GloVe, since it is the state-of-the-art and since our objectives look similar to theirs but has an important difference: we fit the data without context vectors. It is a prior unclear we can do so but our theory implies its validity, so the comparison provides support for the analysis.

6.1 Performance

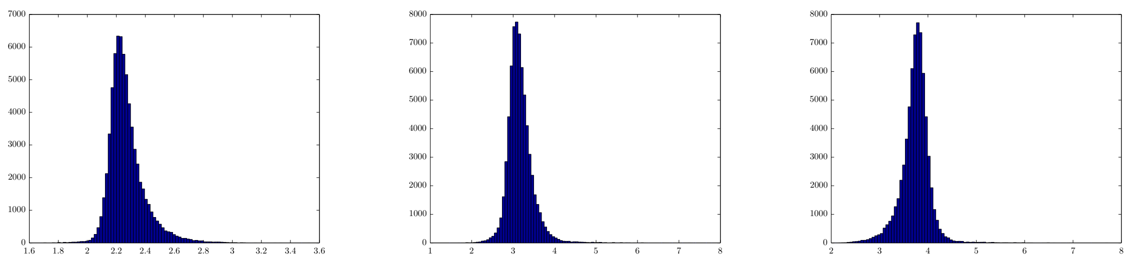
The performance of different methods is presented in Table 1. All our three methods achieve performance comparable to the state-of-the-art approach, especially on semantic tasks. Our methods achieve accuracy 3% lower than the competitor on syntatic tasks in GOOGLE, 8% lower on those in MSR. This is because our model is built explicitly for modeling semantic meanings; some specific features of the syntactic relations are not reflected, *e.g.*, a word "she" can affect the context by a lot and can determine if the next word is "thinks" rather than "think".

The **Mystery 2** described in the introduction stems from the two seemingly contradictory facts that there is high entrywise error in the optimization, and that the best score only have a small margin over the second best. The first can be observed in the training error. For the second fact, we present in Figure 6.1 a scatter plot of the best and second best scores for the questions in GOOGLE when solved by our method **SN**. It can be seen that for a significant portion of the questions, the two scores are close to each other.



(a) **SN** (b) **BIAS** (c) **PMI**

Figure 3: Isotropy property. The figure shows the histogram of $v_w^\top \mathbb{E}\left[v_{w'} v_{w'}^\top\right] v / v_w^\top v_w$ for 1000 random word vector w. x-axis is normalized by the mean of the values.



(a) **SN** (b) **BIAS** (c) **PMI**

Figure 4: Illustration of the norms of the word vectors. The figure shows the histogram of the norms.

6.2 Model verification

We also run experiments to provide positive support for the validity of our assumptions. More precisely, we test assumptions: 1) the isotropy property, 2) the distributions of the norms of the word vectors, and 3) the partition function Z_c is roughly the same for different c.

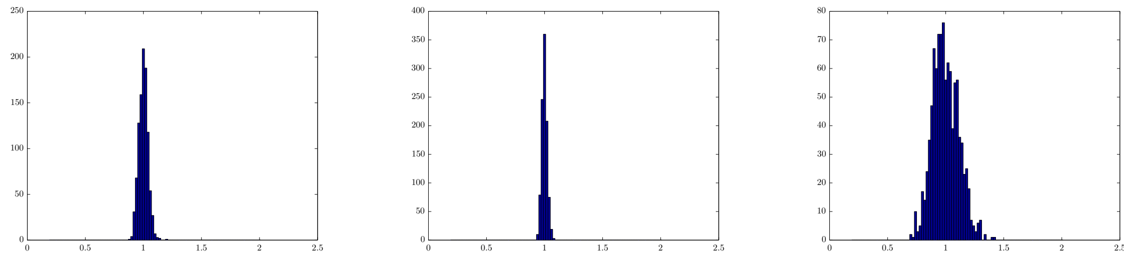
Isotropy We randomly pick a word w, and compute $v_w^\top \mathbb{E}\left[v_{w'} v_{w'}^\top\right] v_w / v_w^\top v_w$. Figure 3 shows the histogram for 1000 random w. For all methods, the values are all reasonably concentrated, mostly in the range $[0.5, 1.5]$ times the mean.

Norm Figure 4 shows the histogram of the norms of the word vectors. It agrees with our assumptions in Section 3: they center around mean with roughly the same standard deviation, and the maximum is bounded by a constant times the mean.

The partition function Z_c. Not knowing the vectors for the contexts, we approximately verified this by computing $Z_c = \sum_{w'} \exp(c^\top w')$ for a random unit vector c. Figure 5 shows the histogram for 1000 random c. The values are concentrated in a range $[0.8, 1.2]$ times the mean.

References

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.



(a) **SN** (b) **BIAS** (c) **PMI**

Figure 5: The partition function Z_c. The figure shows the histogram of Z_c for 1000 random unit vector c. x-axis is normalized by the mean of the values.

David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

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

Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

Geoffrey E Hinton. Distributed representations. 1984.

Omer Levy and Yoav Goldberg. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, Baltimore, Maryland, USA, June. Association for Computational Linguistics*, 2014a.

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

Tomas Mikolov, Stefan Kombrink, Lukas Burget, JH Cernocky, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE, 2011.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013a.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. Citeseer, 2013b.

Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12, 2014.

Douglas L. T. Rohde, Laura M. Gonnerman, and David C. Plaut. An improved model of semantic similarity based on lexical co-occurence. *CACM 2006*, 2006.

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

Supplementary Material: Proofs and Experiment Details

A Proofs of Theorem 1 and 2

In this section we prove Theorem 1 and 2 (restated below) .

Theorem 1 *Assume that the hidden contexts are at stationary distribution, with high probability over the choice of v_w's, we have that for any two different words w and w'*

$$\log p(w, w') = \frac{1}{2d} \|v_w + v_{w'}\|^2 - 2 \log Z \pm o(1) \tag{A.1}$$

for some fixed constant Z. Moreover,

$$\log(p(w)) = \frac{1}{2d} \|v_w\|_2^2 - \log Z \pm o(1). \tag{A.2}$$

Theorem 2 *There exists Z such that for any context c with $|\|c\| - 1| \leq d^{-0.4}$, with high probability $(1 - 2e^{-2n^{0.4}})$) over the choice of v_w's,*

$$(1 - o(1))Z \leq Z_c \leq (1 + o(1))Z.$$

We first prove Theorem 1 using Theorem 2, and Theorem 2 will be proved in Section A.1. For the intuition of the proof, please see Section 4 of the main paper.
Proof [Proof of Theorem 1]
Let c be the hidden context that determines the probability of word w, and c' be the next one that determines w'. We use $p(c'|c)$ to denote the Markov kernel (transition matrix) of the markov chain. Let $\mathcal{C}$ be the stationary distribution of context vector c. We marginalize over the contexts c, c' and then use the independence of w, w' conditioned on c, c',

$$\begin{aligned} p(w, w') &= \int_{c,c'} p(w|c)p(w'|c')p(c, c')dcdc' \\ &= \int_{c,c'} \frac{\exp(\langle v_w, c\rangle)}{Z_c} \frac{\exp(\langle v_{w'}, c'\rangle)}{Z_{c'}} p(c)p(c'|c)dcdc' \end{aligned} \tag{A.3}$$

We first get rid of the partition function Z_c using Theorem 2, which says that there exists Z such that, with probability $1 - 4\exp(-d^{0.2})$,

$$(1 - \epsilon_z)Z \leq Z_c \leq (1 + \epsilon_z)Z. \tag{A.4}$$

where $\epsilon_z = o(1)$.
Let $\mathcal{F}$ be the event that both c and c' satisfy (A.4)and $\overline{\mathcal{F}}$ be its negation, and let $\mathbf{1}_{\mathcal{F}}$ be the indicator function for the event $\mathcal{F}$. Therefore we have $\Pr[\mathcal{F}] \geq 1 - 4\exp(-d^{0.8})$.
We first decompose the integral (A.3) into the two parts according to whether event $\mathcal{F}$ happens,

$$p(w, w') = \int_{c,c'} \frac{1}{Z_c Z_{c'}} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) \mathbf{1}_{\mathcal{F}} dc dc'$$
$$+ \int_{c,c'} \frac{1}{Z_c Z_{c'}} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) \mathbf{1}_{\overline{\mathcal{F}}} dc dc' \tag{A.5}$$

We bound the first quantity on RHS by using (A.4) and the definition of $\mathcal{F}$.

$$\int_{c,c'} \frac{1}{Z_c Z_{c'}} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) \mathbf{1}_{\mathcal{F}} dc dc'$$
$$\leq (1 + \epsilon_z)^2 \frac{1}{Z^2} \int_{c,c'} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) \mathbf{1}_{\mathcal{F}} dc dc' \tag{A.6}$$

and for the second one we use the fact that $Z_c \geq n$ and $\exp(\langle v_w, c\rangle) \leq \exp(2\kappa\sqrt{d})$ (by assumption $\|v_w\| \leq \kappa\sqrt{d}$ and $\|c\| \leq 2$), and conclude

$$\int_{c,c'} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) \mathbf{1}_{\overline{\mathcal{F}}} dc dc'$$
$$\leq \Pr[\overline{\mathcal{F}}] \cdot \exp(4\kappa\sqrt{d}) \leq \exp(-d^{0.7}) \tag{A.7}$$

For the last inequality we use $\Pr[\overline{\mathcal{F}}] \leq 4\exp(-d^{0.2})$. Combining (A.5), (A.6) and (A.7), we obtain

$$p(w, w') \leq (1 + \epsilon_z)^2 \frac{1}{Z^2} \int_{c,c'} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) \mathbf{1}_{\mathcal{F}} dc dc' + \exp(-d^{0.2})$$
$$\leq (1 + \epsilon_z)^2 \frac{1}{Z^2} \left(\int_{c,c'} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) dc dc' + \delta_0 \right)$$

where $\delta_0 = \exp(-d^{0.2}) Z^2 \leq \exp(-d^{0.1})$. This is because $Z \leq \exp(2\kappa) n$ and $d = \omega(\log^2 n)$, and κ is a constant.
On the other hand, we can lowerbound similarly

$$p(w, w') \geq (1 - \epsilon_z)^2 \frac{1}{Z^2} \int_{c,c'} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) \mathbf{1}_{\mathcal{F}} dc dc'$$
$$\geq (1 - \epsilon_z)^2 \frac{1}{Z^2} \left(\int_{c,c'} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) dc dc' - \exp(-d^{0.7}) \right)$$
$$\geq (1 - \epsilon_z)^2 \frac{1}{Z^2} \left(\int_{c,c'} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) dc dc' - \delta_0 \right)$$

Taking logarithm, the multiplicative error translates to a additive error

$$\log p(w, w') = \log \left(\int_{c,c'} \exp(\langle v_w, c\rangle) \exp(\langle v_{w'}, c'\rangle) p(c) p(c'|c) dc dc' \pm \delta_0 \right) - 2\log Z + 2\log(1 \pm \epsilon_z)$$

For the purpose of exploiting the fact that c, c' should be close to each other, we further rewrite $\log p(w, w')$ by re-organizing the integrals,

$$\log p(w,w') = \log\left(\int_c \exp(\langle v_w, c\rangle)p(c)dc \int_{c'} \exp(\langle v_{w'}, c'\rangle)p(c'|c)dc' \pm \delta_0\right) - 2\log Z + 2\log(1\pm\epsilon_z)$$
$$= \log\left(\int_c \exp(\langle v_w, c\rangle)p(c)A(c,c')dc \pm \delta_0\right) - 2\log Z + 2\log(1\pm\epsilon_z) \tag{A.8}$$

where the inner integral which is denoted by $A(c,c')$,

$$A(c,c') := \int_{c'} \exp(\langle v_{w'}, c'\rangle)p(c'|c)dc'$$

Note that by Lemma 3, we have that for any $w \in W$, $\|v_w\|_\infty \le 4\kappa\log n$. Therefore we have that $\langle v_w, c-c'\rangle \le \|v_w\|_\infty\|c-c'\|_1 \le 4\kappa\log n\|c-c'\|_1$.
Then we can bound $A(c,c')$ by

$$\begin{aligned}
A(c,c') &= \int_{c'} \exp(\langle v_{w'}, c'\rangle)p(c'|c)dc' \\
&= \exp(\langle v_{w'}, c\rangle)\int_{c'} \exp(\langle v_{w'}, c'-c\rangle)p(c'|c)dc' \\
&\le \exp(\langle v_{w'}, c\rangle)\int_{c'} \exp(4\kappa|c'-c|_1\log n)p(c'|c) \\
&= \exp(\langle v_{w'}, c\rangle)\mathop{\mathbb{E}}_{p(c'|c)}[\exp(4\kappa|c'-c|_1\log n)] \\
&\le (1+\epsilon_2)\exp(\langle v_{w'}, c\rangle)
\end{aligned}$$

For the lower bound of $A(c,c')$, we first observe that

$$\mathop{\mathbb{E}}_{p(c'|c)}[\exp(4\kappa|c'-c|_1\log n)] + \mathop{\mathbb{E}}_{p(c'|c)}[\exp(-4\kappa|c'-c|_1\log n)] \ge 2$$

Therefore it follows model assumption that

$$\mathop{\mathbb{E}}_{p(c'|c)}[\exp(-4\kappa|c'-c|_1\log n)] \ge 1-\epsilon_2$$

Therefore,

$$\begin{aligned}
A(c,c') &= \exp(\langle v_{w'}, c\rangle)\int_{c'} \exp(\langle v_{w'}, c'-c\rangle)p(c'|c)dc' \\
&\ge \exp(\langle v_{w'}, c\rangle)\int_{c'} \exp(-4\kappa\|c'-c\|\log n)p(c'|c)dc' \\
&= \exp(\langle v_{w'}, c\rangle)\mathop{\mathbb{E}}_{p(c'|c)}[\exp(-4\kappa\|c'-c\|\log n)] \\
&\ge (1-\epsilon_2)\exp(\langle v_{w'}, c\rangle)
\end{aligned}$$

Therefore, we obtain that $A(c,c') = (1\pm\epsilon_2)\exp(\langle v_{w'}, c\rangle)$. Plugging the estimation of $A(c,c')$ into the equation A.8, we obtain that

$$
\begin{aligned}
\log p(w,w') &= \log\left(\int_c \exp(\langle v_w, c\rangle)p(c)A(c,c')dc \pm \delta_0\right) - 2\log Z + 2\log(1\pm\epsilon_z)\\
&= \log\left(\int_c \exp(\langle v_w, c\rangle)p(c)(1\pm\epsilon_2)\exp(\langle v_{w'}, c\rangle)dc \pm \delta_0\right) - 2\log Z + 2\log(1\pm\epsilon_z)\\
&= \log\left(\int_c \exp(\langle v_w, c\rangle)p(c)\exp(\langle v_{w'}, c\rangle)dc \pm \delta_0\right) - 2\log Z + 2\log(1\pm\epsilon_z) + \log(1\pm\epsilon_2)\\
&= \log\left(\int_c \exp(\langle v_w + v_{w'}, c\rangle)p(c)dc \pm \delta_0\right) - 2\log Z + 2\log(1\pm\epsilon_z) + \log(1\pm\epsilon_2)\\
&= \log\left(\mathop{\mathbb{E}}_{c\sim\mathcal{C}}[\exp(\langle v_w + v_{w'}, c\rangle)] \pm \delta_0\right) - 2\log Z + 2\log(1\pm\epsilon_z) + \log(1\pm\epsilon_2)
\end{aligned}
$$

Now it suffices to compute $\mathbb{E}_{c\sim\mathcal{C}}[\exp(\langle v_w + v_{w'}, c\rangle)]$. Let $t = v_w + v_{w'}$. By our assumption, $\mathcal{C}$ is a product distribution across the coordinates. Therefore we can write

$$
\mathop{\mathbb{E}}_{c\sim\mathcal{C}}[\exp(\langle v_w + v_{w'}, c\rangle)] = \prod_{i=1}^{d} \mathop{\mathbb{E}}_{c_i} \exp(t_i c_i)
$$

Using lemma 4 for $t_i c_i \le 1$(we used the fact that $t_i \le 8\kappa \log n$ for all $t = v_w + v_{w'}$, $c_i \le \frac{2}{\sqrt{d}}$ (see Lemma 3); In our setting κ is a constant, $d = \omega((\log n)^2)$), we can estimate $\mathbb{E}_{c_i}\exp(t_i c_i)$ by

$$
\mathop{\mathbb{E}}_{c_i} \exp(t_i c_i) = 1 + \frac{t_i^2}{2d} + O(t_i^4/d^2)
$$

Using the fact that $x - \frac{x^2}{2} \le \ln(1+x) \le x$.

$$
\begin{aligned}
\log \mathop{\mathbb{E}}_{c}[\exp(\langle v_w + v_{w'}, c\rangle)] \quad &= \sum_{i=1}^{d} \log \mathop{\mathbb{E}}_{c_i} \exp(t_i c_i) = \sum_{i=1}^{d} \log\left(1 + \frac{t_i^2}{2d} + O(t_i^4/d^2)\right)\\
&= \sum_{i=1}^{d} \frac{t_i^2}{2d} + O(t_i^4/d^2)\\
&= \|t\|^2/(2d) + O(\|t\|_4^4/d^2)
\end{aligned}
$$

Putting altogether, we have that

$$
\begin{aligned}
\log p(w,w') &= \log\left(\mathop{\mathbb{E}}_{c\sim\mathcal{C}}[\exp(\langle v_w + v_{w'}, c\rangle)] \pm \delta_0\right) - 2\log Z + 2\log(1\pm\epsilon_z) + \log(1\pm\epsilon_2)\\
&= \|v_w + v_{w'}\|^2/(2d) + O(\|v_w + v_{w'}\|_4^4/d^2) + O(\delta_0') - 2\log Z \pm 2\epsilon_z \pm \epsilon_2\\
&= (1+\delta)\|v_w + v_{w'}\|^2/(2d) - 2\log Z \pm 2\epsilon_z \pm \epsilon_2
\end{aligned}
$$

where $\delta_0' = \delta_0 \cdot (\mathbb{E}_{c\sim\mathcal{C}}[\exp(\langle v_w + v_{w'}, c\rangle)])^{-1} = o(1)$ and $\delta = d^{-0.4}$, where in the last step we used Lemma 3 that $\|v_w + v_{w'}\|_4^4/d^2 < \|v_w + v_{w'}\|^2/d^{1.6}$.
Note that ϵ_z, ϵ_2 are on the order of $o(1)$, and $\delta(\|v_w + v_{w'}\|^2/(2d)) = o(1)$ for all w and w' by Lemma 3, we obtain the desired bound,

$$
\log p(w,w') = \frac{1}{2d}\|v_w + v_{w'}\|^2 - 2\log Z \pm o(1)
$$

$\blacksquare$

Lemma 3 *With high probability over the choice of v_w's, we have that for any $w \in W$ and any i, $(v_w)_i \le 4\kappa \log n$, and for any pair of words w, w',*

$$\|v_w + v_{w'}\|_4^4/d^2 < \|v_w + v_{w'}\|^2/d^{1.6}$$

Proof Recall that we assume v_w are generated independently as $v_w = s_w \cdot \hat{v}_w$ where $s_w \le \kappa\sqrt{d}$ for some constant κ and $\hat{v}_w$ is from a spherical Gaussian distribution (each coordinate is i.i.d $N(0, 1/d)$).
Let's do each of the claims separately.
For a standard Gaussian distribution, we know that

$$\Pr\left[|(\hat{v}_w)_i| \ge 4d^{-0.5}\log n\right] \le e^{-\frac{16}{2}\log^2 n}$$

Since $s_w \le \kappa\sqrt{d}$, we know that

$$\Pr\left[|(s_w \cdot \hat{v}_w)_i| \ge 4\kappa \log n\right] \le e^{-\frac{16}{2}\log^2 n}$$

Union bounding, we have that with probability $1 - dne^{-\frac{16}{2}\log^2 n} = 1 - o(1)$ (recall that $d < n^{0.5}$), for all words w, for every coordinate i, $(v_w)_i \le 4\kappa \log n$.
The second claim is not much more difficult. For standard Gaussian distribution, we know that

$$\Pr\left[|(\hat{v}_w)_i| \ge d^{-0.3}/\kappa\right] \le e^{-\frac{d^{0.4}}{2\kappa^2}}$$

Since $s_w \le \kappa\sqrt{d}$, we know that

$$\Pr\left[|(s_w \cdot \hat{v}_w)_i| \ge d^{0.2}\right] \le e^{-\frac{d^{0.4}}{2\kappa^2}}$$

Taking a union bound, we have: with probability $1 - dne^{-\frac{d^{0.4}}{2\kappa^2}}$ (note in our setting $dne^{-\frac{d^{0.4}}{2\kappa^2}} = o(1)$ for large enough d), for all words w and their coordinate i, $|(v_w)_i| \le d^{0.2}$. In this case we have:

$$(v_w + v_{w'})_i^4/d^2 \le (v_w + v_{w'})_i^2/d^{1.6}$$

which easily implies the claim we want.

$\blacksquare$

Lemma 4 *If a real random variable X is symmetric and $\mathbb{E}[X^2] = 1/d$ and $|X| \le 2/\sqrt{d}$ a.s. Then for $t < \sqrt{d}/10$, we have*

$$1 + \frac{t^2}{2d} \le \mathop{\mathbb{E}}_X \exp(tX) \le 1 + \frac{t^2}{2d} + \frac{4}{3}\left(\frac{t}{\sqrt{d}}\right)^4$$

Proof By the moment generating function of X, we have

$$\mathop{\mathbb{E}}_X \exp(tX) = \sum_{j=0}^{\infty} \frac{t^j}{(j)!}\,\mathbb{E}[X^j]$$

Therefore by the assumption that X is symmetric and $\mathbb{E}[X^2] = \frac{1}{d}$, we have that

$$\mathop{\mathbb{E}}_X \exp(tX) \ge 1 + \frac{t^2}{2d}$$

On the other hand, using the fact that $|X| < \frac{2}{\sqrt{d}}$ a.s.

$$\mathop{\mathbb{E}}_{X} \exp(tX) = \sum_{j=0}^{\infty} \frac{t^{2j}}{(2j)!} \mathbb{E}[X^{2j}] = 1 + \frac{t^2}{2d} + \sum_{j=2}^{\infty} \frac{1}{(2j)!} \left(\frac{2t}{\sqrt{d}}\right)^{2j} \leq 1 + \frac{t^2}{2d} + \frac{4}{3}\left(\frac{t}{\sqrt{d}}\right)^4$$

the last inequality is because we choose $t < \sqrt{d}/10$: $\frac{t}{\sqrt{d}} < 1/10$; hence

$$\sum_{j=2}^{\infty} \frac{1}{(2j)!} \left(\frac{2t}{\sqrt{d}}\right)^{2j} \leq \left(\frac{t}{\sqrt{d}}\right)^4 \sum_{j=0}^{\infty} \left(\frac{1}{5}\right)^{2j} \leq \frac{4}{3}\left(\frac{t}{\sqrt{d}}\right)^4$$

■

A.1 Analyzing partition function Z_c

In this section, we prove Theorem 2. We basically first prove that for the means of Z_c are all $(1+o(1))$-close to each other, and then prove that Z_c is concentrated around its mean. It turns out the concentration part is non trivial because the random variable of concern, $\exp(\langle v_w, c\rangle)$ is not well-behaved in terms of the tail. Note that $\exp(\langle v_w, c\rangle)$ is NOT sub-gaussian for any variance proxy. This essentially disallows us to use an existing concentration inequality directly. We get around this issue by considering the truncated version of $\exp(\langle v_w, c\rangle)$, which is bounded, and have similar tail properties as the original one, in the regime that we are concerning.

Proof [Proof of Theorem 2]
Recall that by definition

$$Z_c = \sum_w \exp(\langle v_w, c\rangle).$$

We fix context c and view v_w as random variables throughout this proof. For convenience, we denote the norm of c by $\ell = \|c\|$. Recall that v_w is composed of $v_w = s_w \cdot \hat{v}_w$, where s_w is the scaling and $\hat{v}_w$ is from spherical Gaussian with covariance $\frac{1}{d}I_{d\times d}$ and thus almost a unit vector.
Just as a warm-up, we lowerbound the mean of Z_c as follows:

$$\mathbb{E}[Z_c] = n\,\mathbb{E}\left[\exp(\langle v_w, c\rangle)\right] \geq n\,\mathbb{E}\left[1 + \langle v_w, c\rangle\right] = n$$

On the other hand, to upperbound the mean of Z_c, we condition on the scaling s_w,

$$\begin{aligned}\mathbb{E}[Z_c] &= n\,\mathbb{E}[\exp(\langle v_w, \tilde{v}_c\rangle)] \\ &= n\,\mathbb{E}\left[\mathbb{E}\left[\exp(\langle v_w, \tilde{v}_c\rangle) \mid s_w\right]\right]\end{aligned}$$

Note that conditioned on s_w, we have that $\langle v_w, \tilde{v}_c\rangle$ is a guassian random variable with variance $\sigma^2 = \|c\|^2 s_w^2/d$. Therefore,

$$\begin{aligned}\mathbb{E}\left[\exp(\langle v_w, \tilde{v}_c\rangle) \mid s_w\right] &= \int_x \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2}) \exp(x) dx \\ &= \int_x \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\sigma^2)^2}{2\sigma^2} + \sigma^2/2) dx \\ &= \exp(\sigma^2/2)\end{aligned}$$

It follows that

$$\mathbb{E}[Z_c] = \mathbb{E}[\exp(\sigma^2/2)] = \mathbb{E}[\exp(s_w^2\|c\|^2/2)] = \mathbb{E}[\exp(s^2\|c\|^2/2)]$$

Let $Z := \mathbb{E}[\exp(|s|^2/2d)]$. By Proposition 5, we have that $1 - o(d^{-0.4}) \le \|c\| \le 1 + o(d^{-0.4})$. Therefore, for any c,

$$\mathbb{E}[Z_c] = \mathbb{E}[\exp(s^2\|c\|^2/2d)] \le \mathbb{E}[\exp(s^2/2)\cdot \exp(o(d^{-0.4})s^2/2d)] \le (1+o(d^{-0.4})\kappa^2/2)Z = (1+o(1))Z$$

Similarly we can prove that $\mathbb{E}[Z_c] \ge (1-o(1))Z$ for any c.
We calculate the variance of Z_c as follows:

$$\begin{aligned}\mathbb{V}[(Z_c - \mathbb{E}Z_c)^2] &= \sum_w \mathbb{V}\left[\exp(\langle v_w, c\rangle)^2\right] \le n\,\mathbb{E}[\exp(2\langle v_w, c\rangle)] \\ &= n\,\mathbb{E}\left[\mathbb{E}\left[\exp(2\langle v_w, c\rangle) \mid s_w\right]\right]\end{aligned}$$

By a very similar calculation as above, using the fact that $\langle v_w, c\rangle$ is a guassian random variable with variance $\sigma^2 = \ell^2\|s_w\|^2/d$,

$$\begin{aligned}\mathbb{E}\left[\exp(\langle v_w, c\rangle^2) \mid s_w\right] &= \int_x \frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{x^2}{2\sigma^2})\exp(2x)dx \\ &= \int_x \frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(x-2\sigma^2)^2}{2\sigma^2} + 2\sigma^2)dx \\ &= \exp(2\sigma^2)\end{aligned}$$

Therefore, we have that

$$\begin{aligned}\mathbb{E}[(Z_c - \mathbb{E}Z_c)^2] &\le n\,\mathbb{E}\left[\mathbb{E}\left[\exp(2\langle v_w, \tilde{v}_c\rangle) \mid s_w\right]\right] \\ &= n\,\mathbb{E}\left[\exp(2\sigma^2)\right] = n\,\mathbb{E}\left[\exp(2\ell^2\|s_w\|^2/d)\right] \le \Lambda n\end{aligned}$$

For $\Lambda = \exp(8\kappa^2)$ being a constant. Therefore, the standard deviation of Z_c is $\sqrt{\Lambda n}$ is much less than n. Also note that $\mathbb{E}[Z_c] \ge n$, therefore we should expect with good probability over the choice of v_w's, we have that Z_c is within $\mathbb{E}[Z_c] \pm \sqrt{\Lambda n} = \mathbb{E}[Z_c](1+o(1))$.
However, observe that $\exp(\langle v_w, c\rangle)$ is not sub-Gaussian or bounded. This disallows us to apply the usual concentration inequalities. The rest of the proof deals with this issue in a slightly more specialized manner.
Let's define $\mathcal{F}_w$ be the event that $\exp(\langle v_w, c\rangle) < d^{0.2}$. Observe that $\mathcal{F}$ is a very high probability event with $\Pr[\mathcal{F}_w] \ge 1 - \exp(-d^{0.2}/\kappa^2)$. Let random variable X_w have the same distribution as $\exp(\langle v_w, c\rangle)|_{\mathcal{F}_w}$.
We prove concentration inequality for $Z'_c = \sum_w X_w$. Observe that mean of Z'_c is lowerbounded

$$\mathbb{E}[Z'_c] = n\,\mathbb{E}\left[\exp(\langle v_w, c\rangle)|_{\mathcal{F}_w}\right] \ge n\exp(\mathbb{E}\left[\langle v_w, c\rangle|_{\mathcal{F}_w}\right]) = n$$

and the variance is upperbounded by

$$\begin{aligned}\mathbb{V}[Z'_c] &\le n\,\mathbb{E}\left[\exp(\langle v_w, c\rangle)^2|_{\mathcal{F}_w}\right] \\ &\le \frac{1}{\Pr[\mathcal{F}_w]}\mathbb{E}\left[\exp(\langle v_w, c\rangle)^2\right] \\ &\le \frac{1}{\Pr[\mathcal{F}_w]}\Lambda n \le 1.1\Lambda n\end{aligned}$$

where the second line uses the fact that

$$\begin{aligned}\mathbb{E}\left[\exp(\langle v_w, c\rangle)^2\right] \quad &= \Pr[\mathcal{F}_w]\,\mathbb{E}\left[\exp(\langle v_w, c\rangle)^2|\mathcal{F}_w\right] + \Pr[\overline{\mathcal{F}}_w]\,\mathbb{E}\left[\exp(\langle v_w, c\rangle)^2|\overline{\mathcal{F}}_w\right] \\ &\ge Pr[\mathcal{F}_w]\,\mathbb{E}\left[\exp(\langle v_w, c\rangle)^2|\mathcal{F}_w\right].\end{aligned}$$

Moreover, by definition, for any w, $|X_w| \le d^{0.2}$. Therefore by Bernstein's inequality, we have that

$$\Pr\left[|Z'_c - \mathbb{E}[Z'_c]| > 4\sqrt{\Lambda n} + 12n^{0.7}\right] \le e^{-2n^{0.4}}$$

By the fact that $\mathbb{E}[Z'_c] \ge n$, we have that for $\epsilon = n^{-0.3} \le d^{-0.6}$ (we use the fact that $d < n^{0.5}$)

$$\Pr\left[|Z'_c - \mathbb{E}[Z'_c]| > \epsilon\,\mathbb{E}[Z'_c]\right] \le 2e^{-2n^{0.4}}$$

Let $\mathcal{F} = \cap_w \mathcal{F}_w$ be the union of all $\mathcal{F}_w$. We have that by definition, Z'_c have the same distribution as $Z_c|_{\mathcal{F}}$. Therefore, we have that

$$\Pr[|Z_c - \mathbb{E}[Z'_c]| > \epsilon\,\mathbb{E}[Z'_c] \mid \mathcal{F}] \le 2e^{-2n^{0.4}}$$

and therefore

$$\Pr[|Z_c - \mathbb{E}[Z'_c]| > \epsilon\,\mathbb{E}[Z'_c]] \le \frac{1}{\Pr[\mathcal{F}]} \cdot \Pr[|Z_c - \mathbb{E}[Z'_c]| > \epsilon\,\mathbb{E}[Z'_c] \mid \mathcal{F}] \le 2e^{-2n^{0.4}}$$

Finally we show that $\mathbb{E}[Z'_c]$ are close to each other as well. We take c that satisfies that $\|c\| = 1 \pm d^{-0.4}$ and consider $\mathbb{E}[Z_c] = \mathbb{E}[\exp(\langle v, c\rangle)|\mathcal{F}]$, where v is from the same distribution where v_w is generated, and $\mathcal{F}$ is the event that $\exp(\langle v, c\rangle) \le d^{0.2}$. Note that random variable $\exp(\langle v, c\rangle)|\mathcal{F}$ is really rational invariant with respect to c. Therefore we have that $\mathbb{E}[Z'_c] = \mathbb{E}[\exp(\langle v, \|c\|z\rangle)|\mathcal{F}]$, where z is any unit vector in the space, and $\mathcal{F}'$ is the event that $\exp(\langle v, z\rangle) \le d^{0.2}$.

$$\begin{aligned}
\mathbb{E}[Z'_{c_1}] \quad & \le \mathbb{E}[\exp(\langle v, \|c\|z\rangle)|\mathcal{F}'] \le \mathbb{E}[\exp(\langle v, z\rangle)|\mathcal{F}'] \sup\{\exp(\langle v, (\|c\|-1)z\rangle)|\mathcal{F}'\} && \text{(A.9)}\\
& = \mathbb{E}[\exp(\langle v, \|c\|z\rangle)|\mathcal{F}'] \exp(d^{-0.2}) && \text{(A.10)}\\
& = (1 + o(1))\,\mathbb{E}[\exp(\langle v, \|c\|z\rangle)|\mathcal{F}'] && \text{(A.11)}
\end{aligned}$$

Similarly we can prove that

$$\mathbb{E}[Z'_{c_1}] \quad \ge (1 - o(1))\,\mathbb{E}[\exp(\langle v, \|c\|z\rangle)|\mathcal{F}'] \tag{A.12}$$

Therefore, let $Z = \mathbb{E}[\exp(\langle v, \|c\|z\rangle)|\mathcal{F}']$, we have the desired result.

■

Proposition 5 *When $c \sim \mathcal{C}$ is at stationary distribution of the random walk, we have that*

$$\Pr_{c\sim\mathcal{C}}\left[|\|c\| - 1| > 2d^{-0.4}\right] \le 2\exp(-d^{0.2})$$

Proof By assumption, each coordinate of c is independent with $\mathbb{E}[c_i^2] = \frac{1}{d}$ and $|c_i|^2 \le \frac{4}{d}$, so the Proposition 5 follows from standard Chernoff bound. ■

B Maximum Likelihood Estimator for log of Binomial Distribution

In this section, we present a very simple calculation that show that the weight function $f_{w,w'}$ in GloVe makes sense. Concretely, we assume that the co-occurrence count $X_{w,w'}$ for words pair w, w' is from binomial distribution $\mathsf{bin}(|T|, p(w, w'))$. We show that the likelihood of $\log X_{w,w'}$ given on $p(w, w') = x/|T|$ is of the form

$$C + \frac{X_{w,w'}}{|T| - X_{w,w'}} (x - \log(X_{w,w'}))^2 + \text{high order term}$$

where C is a fixed constant (that depends on the data $X_{w,w'}$ but not x or p.)

Theorem 6 *Suppose X is from binomial distribution* $\mathsf{bin}(m,p)$ *with $p = x/m$, then we have that the likelihood of X is of the form*

$$\Pr[X \mid p] = C + \frac{X}{m - X}\,(x - \log(X))^2 + O\left((x - \log(X))^3\right)$$

where $x = \log(pm)$, and C only depends on X but not p or x.

Proof
Let $X \sim \mathsf{bin}(m,p)$, then we know that the probability mass is $\Pr[X = k \mid p] = \binom{m}{k}(1-p)^{m-k}p^k, k \in \{0, 1, ..., m\}$.
Therefore, the log-likelihood is given by:

$$\log \Pr[X = k \mid p] = \log \binom{m}{k} + (m - k)\log(1 - p) + k \log p, k \in \mathbb{N}$$

Let $x = \log(mp)$, we have: for $k \in \mathbb{N}$,

$$\log \Pr[X = k \mid p] = \log \binom{m}{k} + (m - k)\log\left(1 - \frac{e^x}{m}\right) + k \log \frac{e^x}{m}$$

We take the Taylor expansion w.r.t x at point $\log k$, and conclude

$$\log \Pr[X = k \mid p] = \log \binom{m}{k} - \frac{km}{2(m-k)}(x - \log k)^2 + mO\left((x - \log k)^3\right)$$

$\blacksquare$

C Experiment Details

Training method The data set used is the English Wikipedia [1]. We preprocessed the data by standard approach (removing non-textual elements, sentence splitting, and tokenization) [2], resulting in a data set with about 3 billion tokens. We ignored words that appeared less than 1000 times in the corpus and obtained a vocabulary of 68,430. The co-occurrence is then computed by using GloVe's code, using a window size of 10. All the methods are then trained on this co-occurrence for fair comparison. For all our objectives, we use AdaGrad (**?**) for the optimization with initial learning rate of 0.05, and run 100 iterations for all our objectives.

Evaluation method The constructed vectors are evaluated on two standard testbeds for word analogy tasks (GOOGLE[3], MSR[4]) and also on a more challenging testbed we collected. The word analogy task consists of questions like, "a:b::c:??." The algorithm should return a list of candidates. The GOOGLE and MSR questions are answered correctly only if the correct missing term is ranked top 1. For those more difficult questions in our testbed, they are answered correctly if the missing term is among the top 10. The GOOGLE testbed contains 19,544 such questions, including a semantic subset (7874 questions divided into 5 types) and a syntactic subset (10167 questions divided into 9 types). A typical semantic question is "man:woman::king:?" and a syntactic one is "run:runs::walk:?." The MSR includes 8000 syntactic questions for adjectives, nouns and verbs. Our testbed includes 180 semantic questions collected from English courses, such as "lettuce:vegetable::apples:?." This testbed will be released with the public version of the paper.

[1] http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2
[2] We used the script provided by Matt Mahoney. The script is at the bottom of http://mattmahoney.net/dc/textdata.html.
[3] code.google.com/p/word2vec/source/browse/trunk/questions-words.txt
[4] research.microsoft.com/en-us/projects/rnn/

	Relations	GloVe	**SN**	**BIAS**	**PMI**
GOOGLE	capital-common-countries	96.44	96.25	97.04	96.05
	capital-world	97.07	94.17	96.01	94.15
	currency	8.65	8.65	8.17	8.65
	city-in-state	73.33	67.29	72.8	66.88
	family	90.00	89.76	88.57	89.76
	gram1-adjective-to-adverb	28.33	23.08	21.17	23.39
	gram2-opposite	35.98	35.71	36.24	35.19
	gram3-comparative	84.68	84.53	84.68	84.38
	gram4-superlative	54.37	43.68	45.86	42.87
	gram5-present-participle	62.69	56.34	58.52	56.44
	gram6-nationality-adjective	92.06	90.87	91.68	90.87
	gram7-past-tense	53.33	49.81	48.91	49.87
	gram8-plural	83.36	79.08	82.69	79.58
	gram9-plural-verbs	56.40	54.93	52.59	55.67
	semantic	84.54	81.13	83.77	80.98
	syntactic	64.62	61.15	61.66	61.19
	total	73.32	69.87	71.31	69.82
MSR	adjective	54.01	50.00	51.81	49.24
	noun	73.10	69.70	70.50	68.60
	verb	59.43	47.70	48.73	48.33
	total	61.01	52.54	53.68	52.54
ours	total(@top10)	35.09	35.67	35.09	30.41

Table 1: Accuracy on the word analogy tasks.

Our vocabulary covers about 92% of the GOOGLE questions, 63% of MSR, and 92% of our testbed. To solve these tasks, we use linear algebraic query, *i.e.*, rank d according to $\langle v_d, v_c + v_b - v_a \rangle$. We then compare our performance with GloVe[5], trained with the following command:

```
./glove -save-file $SAVE_FILE -threads 8 -input-file $COOCCURRENCE_SHUF_FILE -x-max 100
-iter 1000 -vector-size 300 -binary 2 -vocab-file $VOCAB_FILE -verbose 2 -model 0
```

C.1 Performance

The performance of different methods is presented in Table 1. All our three methods achieve performance comparable to the state-of-the-art approach, especially on semantic tasks. On syntatic tasks, our methods achieve accuracy 3% lower than the competitor. This is because our model is built explicitly for modeling semantic meanings; some specific features of the syntactic relations are not reflected, *e.g.*, a word "she" can affect the context by a lot and can determine if the next word is "thinks" rather than "think".

The **Mystery 2** described in the introduction stems from the two seemingly contradictory facts that there is high entrywise error in the optimization, and that the best score only have a small margin over the second best. The first can be observed in the training error. For the second fact, we present in Figure C.1 a scatter plot of the best and second best scores for the questions in GOOGLE when solved by our method **SN**. It can be seen that for a significant portion of the questions, the two scores are close to each other.

C.2 Model verification

We also run experiments to test some assumptions in our model. The results agree with our model and analysis, providing positive support for the validity of our assumptions. More precisely, we test three key

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

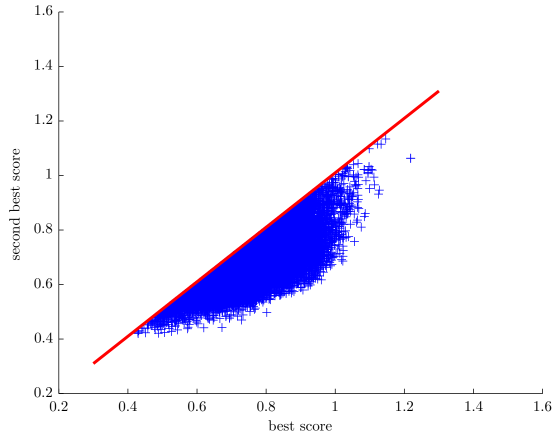


Figure 1: The scatter plot showing the small gap between the best score and the second best score for all questions in the testbed GOOGLE. The red line is $x = y$.

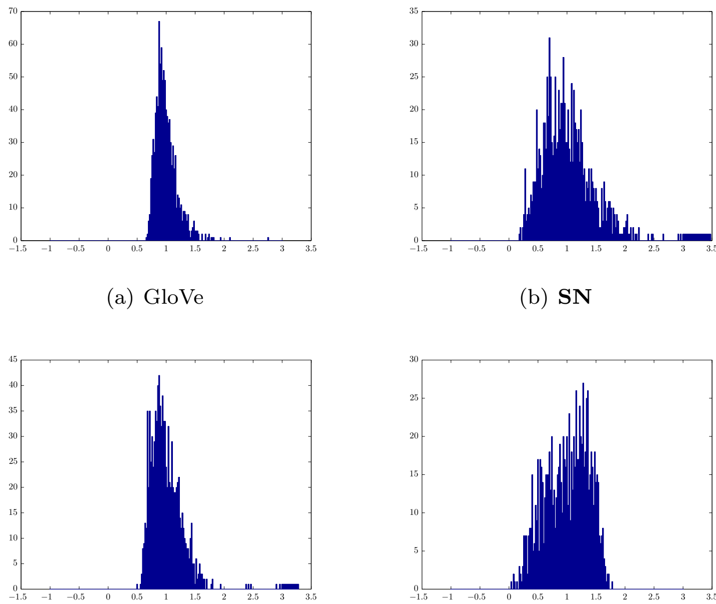
elements in our model: the isotropy property, the distributions of the norms and directions of the word vectors. Furthermore, we also test a key technical result followed from these assumptions that the partition function Z_c is roughly the same for different c.

Isotropy We randomly pick a word w, and compute $v_w^\top \mathbb{E}\left[v_{w'} v_{w'}^\top\right] v_w / v_w^\top v_w$. Figure 2 shows the histogram for 1000 random w. It can be observed that for all methods, the values are all reasonably concentrated, mostly in the range $[0.5, 1.5]$ times the mean.

Norm Figure 3 shows the histogram of the norms of the word vectors. It agrees with our assumptions: they center around mean with roughly the same standard deviation, and the maximum is bounded by a constant times the mean.

Direction To check the directions of the vectors, we randomly sample 1000 unit vector v, and compute $v^\top \mathbb{E}\left[v_{w'} v_{w'}^\top\right] v$. Figure C.2 is the histogram of these values, which shows that they are tightly concentrated around the mean, which agrees with our assumption that the direction is roughly uniform over space.

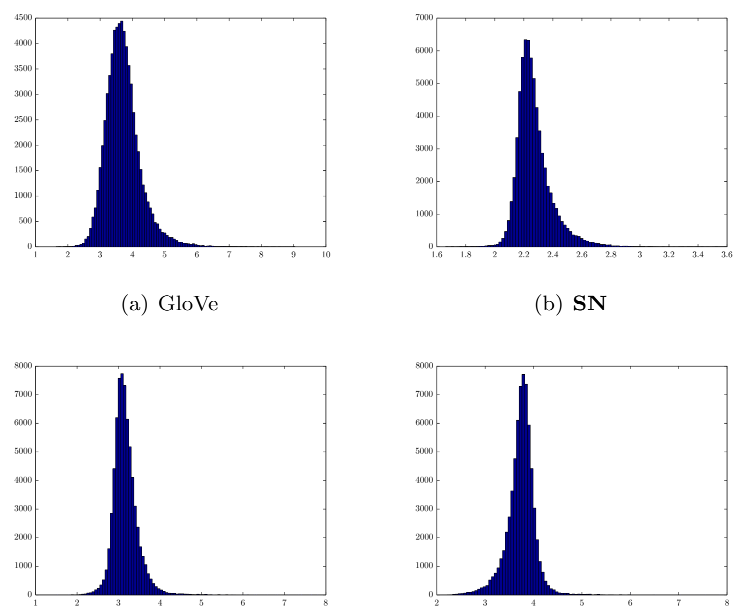
The partition function Z_c Note that we do not know the vectors for the contexts, so we approximately verified this by computing $Z_c = \sum_{w'} \exp(c^\top w')$ for a random unit vector c. Figure 5 shows the histogram for 1000 random c. The values are concentrated, mostly in the range $[0.8, 1.2]$ times the mean.



(a) GloVe (b) **SN**

(c) **BIAS** (d) **PMI**

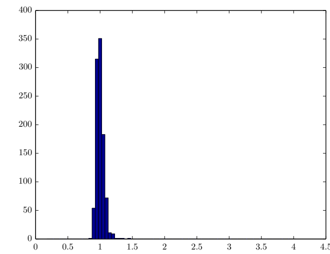
Figure 2: Isotropy property. The figure shows the histogram of $v_w^\top \mathbb{E}\left[v_{w'} v_{w'}^\top\right] v / v_w^\top v_w$ for 1000 random word vector w. x-axis is normalized by the mean of the values.



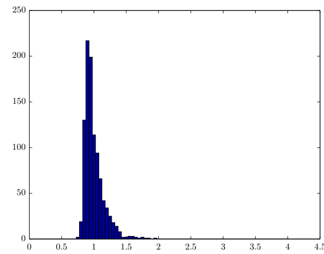
(a) GloVe (b) **SN**

(c) **BIAS** (d) **PMI**

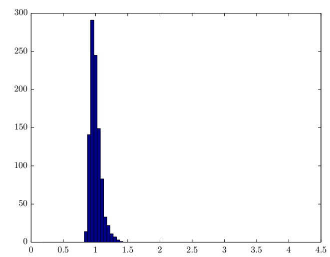
Figure 3: Illustration of the norms of the word vectors. The figure shows the histogram of the norms.



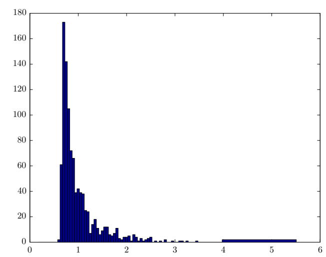
(a) GloVe



(b) **SN**

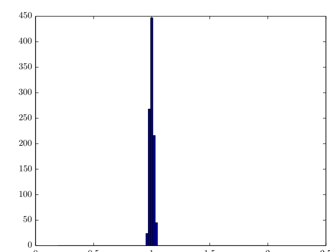


(c) **BIAS**

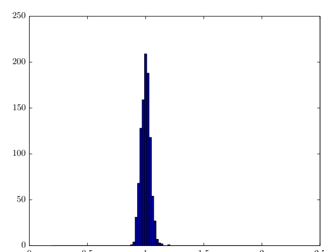


(d) **PMI**

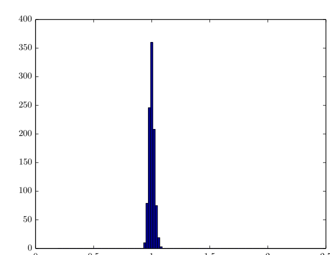
Figure 4: Illustration of the directions of the word vectors. The figure shows the histogram of $v^\top \mathbb{E}\left[v_{w'} v_{w'}^\top\right] v$ for 1000 random unit vector v. x-axis is normalized by the mean of the values.



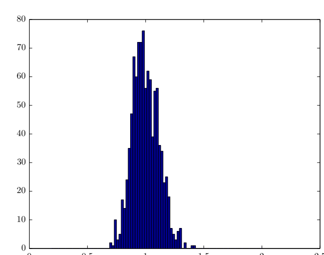
(a) GloVe



(b) **SN**



(c) **BIAS**



(d) **PMI**

Figure 5: The partition function Z_c. The figure shows the histogram of Z_c for 1000 random unit vector c. x-axis is normalized by the mean of the values.