Journal of Artificial Intelligence Research 66 (2019) 225-242 Submitted 02/2019; published 09/2019

Context Vectors Are Reflections of Word Vectors in Half the Dimensions

Zhenisbek Assylbekov ZHASSYLBEKOV@NU.EDU.KZ
Rustem Takhanov RUSTEM.TAKHANOV@NU.EDU.KZ
Nazarbayev University, Department of Mathematics,
53 Kabanbay Batyr ave., Astana 010000 Kazakhstan

Abstract

This paper takes a step towards the theoretical analysis of the relationship between word embeddings and context embeddings in models such as word2vec. We start from basic probabilistic assumptions on the nature of word vectors, context vectors, and text generation. These assumptions are supported either empirically or theoretically by the existing literature. Next, we show that under these assumptions the widely-used word-word PMI matrix is approximately a random symmetric Gaussian ensemble. This, in turn, implies that context vectors are reflections of word vectors in approximately half the dimensions. As a direct application of our result, we suggest a theoretically grounded way of tying weights in the SGNS model.[1]

1. Introduction and Main Result

Today word embeddings play an important role in many natural language processing tasks, from predictive language models and machine translation to image annotation and question answering, where they are usually plugged into a larger model. An understanding of their properties is of interest as it may allow the development of embeddings that are better both in interpretability and quality of models built upon them. This paper takes a step in this direction.

Notation: We let $\mathbb{R}$ denote the real numbers. Bold-faced lowercase letters ($\mathbf{x}$) denote vectors in Euclidean space, bold-faced uppercase letters ($\mathbf{X}$) denote matrices, plain-faced lowercase letters (x) denote scalars, plain-faced uppercase letters (X) denote scalar random variables, $\|\cdot\|$ denotes the Euclidean norm: $\|\mathbf{x}\| := \sqrt{\mathbf{x}^\top \mathbf{x}}$, 'i.i.d.' stands for 'independent and identically distributed'. We use the sign $\sim$ to abbreviate the phrase 'distributed as', and the sign $\propto$ to abbreviate 'proportional to'. $\mathrm{Tr}(\mathbf{A})$ is used to denote the trace of a matrix $\mathbf{A}$. $M_{\mathbf{x}}(\mathbf{t})$ is the moment-generating function of a random vector $\mathbf{x}$ at $\mathbf{t}$: $M_{\mathbf{x}}(\mathbf{t}) = \mathbb{E}[e^{\mathbf{t}^\top \mathbf{x}}]$. $\odot$ is the Hadamard product (element-wise multiplication). $\mathbf{A}_{a:b,c:d}$ is a submatrix located at the intersection of rows $a, a+1, \ldots, b$ and columns $c, c+1, \ldots, d$ of a matrix $\mathbf{A}$.

Assuming that words have already been converted into indices, let $\{1, \ldots, n\}$ be a finite vocabulary of words. Following the setup of the widely used WORD2VEC model (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), we will use *two* vectors per each word i:

- $\mathbf{w}_i$ is an embedding of the word i when i is a center word,
- $\mathbf{c}_i$ is an embedding of the word i when i is a context word.

1. Our modification of the SGNS is available at `https://github.com/zh3nis/word2vec_wt`

We make the following key assumptions in our work.

Assumption 1. *A priori word vectors* $\mathbf{w}_1, \dots, \mathbf{w}_n \in \mathbb{R}^d$ *are i.i.d. draws from isotropic multivariate Gaussian distribution:*

$$\mathbf{w}_i \overset{iid}{\sim} \mathcal{N}\left(\mathbf{0},\, \tfrac{1}{d}\mathbf{I}\right), \tag{1}$$

where $\mathbf{I}$ *is the* $d \times d$ *identity matrix.*

This is motivated by the work of Arora, Li, Liang, Ma, and Risteski (2016), where the ensemble of word vectors consists of i.i.d. draws generated by $\mathbf{v} = s \cdot \hat{\mathbf{v}}$, with $\hat{\mathbf{v}}$ being from the spherical Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and s being a scalar random variable with bounded expectation and range. In their work, the norm $\|\mathbf{v}_i\|$ of the word vector for a word i is related to its unigram probability $p(i)$, and to allow a sufficient dynamic range for these probabilities they needed the multiplier s. In our work, unigram probabilities are not mapped to vector lengths, and this is why we do not need such multiplier. Direct relationship between word probabilities and word vector norms is also implied by the model of Hashimoto, Alvarez-Melis, and Jaakkola (2016).

Assumption 2. *Context vectors* $\mathbf{c}_1, \dots, \mathbf{c}_n$ *are related to word vectors according to*

$$\mathbf{c}_i = \mathbf{Q}\mathbf{w}_i, \quad i = 1, \dots, n, \tag{2}$$

for some orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$.

This is mainly guided by the work of Press and Wolf (2017), who showed that context vectors in the SGNS model of Mikolov et al. (2013) are distributed similarly to word vectors in the sense that pairwise cosine distances between word (input) embeddings strongly correlate with the corresponding pairwise cosine distances between context (output) embeddings (see their Table 4). This is why we choose the transform from word vectors to context vectors to be orthogonal as it preserves inner products and consequently Euclidean norms. Notice, that $\mathbf{c}_i \overset{\text{iid}}{\sim} \mathcal{N}\left(\mathbf{0}, \tfrac{1}{d}\mathbf{I}\right)$.

Assumption 3. *Given a word* j*, the probability of any word* i *being in its context*[2] *is given by*

$$p(i \mid j) \propto p_i \cdot e^{\mathbf{w}_j^\top \mathbf{c}_i} \tag{3}$$

where $p_i = p(i)$ *is the unigram probability for the word* i*, which is inverse proportional to its smoothed frequency rank* r_i*, i.e.*

$$p_i \propto \frac{1}{r_i^{1-\alpha}}, \qquad \alpha \in (0, 1]. \tag{4}$$

This is similar to the log-linear model of Arora et al. (2016), but differs in the following aspects: $\mathbf{c}_i$ is not assumed to do a random walk over the unit sphere with bounded displacement; we use the factor p_i to directly capture word frequencies and do not model them via vector norms. Equation (3) can be interpreted as follows: probability that the word i

2. Context is a fixed-size symmetric window around the given word.

occurs in the context of the word j is the probability that the word i occurs anywhere in a large corpus, corrected for the relationship between words i and j. This approach was already considered by Melamud, Dagan, and Goldberger (2017) but in their work i is the entire left context of the word j, and $\mathbf{c}_i$ is a vector representation of this entire context. Also, like Arora et al. (2016) but unlike Melamud et al. (2017), we use the model (3) for a theoretical analysis rather than for fitting to data. Smoothing of the unigram probabilities (i.e. raising them to power $1-\alpha$) is motivated by the works of Mikolov et al. (2013), Levy, Goldberg, and Dagan (2015), Pennington, Socher, and Manning (2014), where $\alpha = 0.25$ is a typical choice. We notice here that $\alpha = 0^+$ gives us Zipf's law (Zipf, 1935), whereas $\alpha = 1$ gives us uniform distribution of word frequencies which is not valid empirically but on the other hand can be used to explain additivity of word vectors (Gittens, Achlioptas, & Mahoney, 2017).

The relationship between word (input) and context (output) vectors was addressed in several previous works. E.g., in recurrent neural network language modeling (RNNLM), tying input and output embeddings is a useful regularization technique introduced earlier (Bengio, Ducharme, Vincent, & Jauvin, 2001) and studied in more details recently (Press & Wolf, 2017; Inan, Khosravi, & Socher, 2017). This technique improves language modeling quality (measured as perplexity of a held-out text) while decreasing the total number of trainable parameters almost two-fold since most of the parameters in RNNLM are due to embedding matrices. The direct application of this regularization technique to SGNS worsens the quality of word vectors as was shown empirically by Press and Wolf (2017) and by Gulordava, Aina, and Boleda (2018). This worsening was predicted earlier by Goldberg and Levy (2014) using a simple linguistic observation that words usually do not appear in the contexts of themselves. This basically means that $\mathbf{Q} \neq \mathbf{I}$ in (2). At the same time, there is empirical evidence that the relationship between input and output embeddings *is* linear (Mimno & Thompson, 2017; Gulordava et al., 2018). In this paper, we provide a theoretical justification for this and reveal the exact form of the transform $\mathbf{Q}$. Our main contribution is the following

Theorem 1. *Under Assumptions 1, 2, and 3 above, the context vector $\mathbf{c}_i$ for a word i is a reflection of the word vector $\mathbf{w}_i$ in approximately half of the dimensions.*

Figure 1 illustrates this idea for the case $d = 2$. In general, our word and context vectors live in a d-dimensional vector space over real numbers ($\mathbb{R}^d$). By Theorem 1 we can settle them in a $d/2$-dimensional vector space over complex numbers ($\mathbb{C}^{d/2}$) in such way that the context vector $\widetilde{\mathbf{c}}_i \in \mathbb{C}^{d/2}$ for a word i is the *complex conjugate* of the word vector $\widetilde{\mathbf{w}}_i \in \mathbb{C}^{d/2}$. This is in line with the results of Allen, Balažević, and Hospedales (2018), however they use a completely different set of basic assumptions and their primary goal is to encode statistical properties of words directly into word vectors.

2. Proof of Theorem 1

The proof is divided into three steps: first we show that the partition function in (3) concentrates around 1, and thus $\propto$ can be replaced by $\approx$; using this fact we show that $\mathbf{Q}$ is (approximately) an involutary matrix, i.e. similar to $\mathrm{diag}(\mathbf{q})$, $\mathbf{q} \in \{+1, -1\}^d$; and finally we show that the word-word pointwise-mutual information matrix is approximately a symmet-

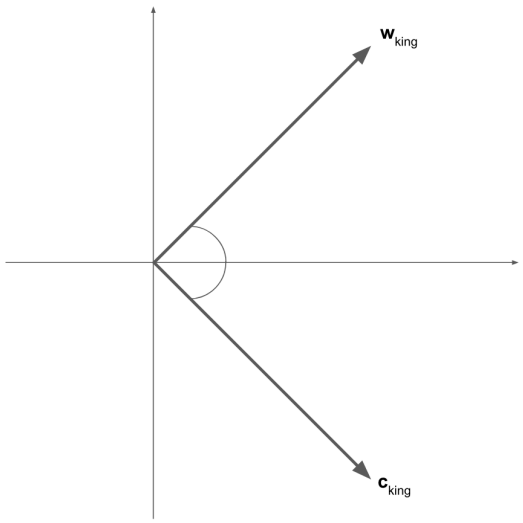

Figure 1: Context vector is a reflection of word vector in half the coordinates.

ric Gaussian random matrix with weakly dependent entries. The latter fact immediately implies the statement of the Theorem 1.

2.1 Concentration of the Partition Function

We first need the following auxiliary result.

Lemma 1. *Let* $\mathbf{w} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$*, then* $\forall t > 0$ *and any orthogonal* $\mathbf{Q}$

$$\Pr\left(\left|\mathbf{w}^\top\mathbf{Q}\mathbf{w} - \mathrm{Tr}(\mathbf{Q})\sigma^2\right| > t\sqrt{2d}\sigma^2\right) \leq \frac{1}{t^2}. \tag{5}$$

Proof. Consider the random variable $X = \mathbf{w}^\top\mathbf{Q}\mathbf{w}$, and let $\mathbf{Q} = [q_{ij}]$. We have

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i,\,j} q_{ij}w_iw_j\right] = \sum_i q_{ii}\sigma^2 = \mathrm{Tr}(\mathbf{Q})\sigma^2. \tag{6}$$

Therefore,

$$(\mathbb{E}[X])^2 = \sum_{i,\,j} q_{ii}q_{jj}\sigma^4. \tag{7}$$

Further,

$$\mathbb{E}\left[X^2\right] = \mathbb{E}\left[\left(\sum_{i,\,j} q_{ij}w_iw_j\right)^2\right] = \mathbb{E}\left[\sum_{i,\,j} q_{ii}q_{jj}w_i^2w_j^2 + \sum_{i\neq j}(q_{ij}^2 + q_{ij}q_{ji})w_i^2w_j^2\right],$$

where we dropped the terms containing odd powers of w_i, as their expectations are zeros. Hence,

$$\begin{aligned}\mathbb{E}\left[X^2\right] &= \sum_{i\neq j} q_{ii}q_{jj}\sigma^4 + \sum_i q_{ii}^2\,\mathbb{E}[w_i^4] + \sum_{i\neq j}(q_{ij}^2 + q_{ij}q_{ji})\sigma^4 \\ &= \sum_{i\neq j} q_{ii}q_{jj}\sigma^4 + 3\sum_i q_{ii}^2\sigma^4 + \sum_{i\neq j}(q_{ij}^2 + q_{ij}q_{ji})\sigma^4 \\ &= \sum_{i,j} q_{ii}q_{jj}\sigma^4 + 2\sum_i q_{ii}^2\sigma^4 + \sum_{i\neq j}(q_{ij}^2 + q_{ij}q_{ji})\sigma^4. \end{aligned} \tag{8}$$

From (7) and (8) we have

$$\mathrm{Var}[X] = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2 = 2\sum_i q_{ii}^2\sigma^4 + \sum_{i\neq j}(q_{ij}^2 + q_{ij}q_{ji})\sigma^4.$$

It is easy to see that $\sum_i q_{ii}^2 + \sum_{i\neq j} q_{ij}^2 = d$ (the sum of squared elements of an orthogonal matrix). In this way,

$$\mathrm{Var}[X] = \left(d + \sum_i q_{ii}^2 + \sum_{i\neq j} q_{ij}q_{ji}\right)\sigma^4 = \left(d + \mathrm{Tr}(\mathbf{Q}^2)\right)\sigma^4. \tag{9}$$

Applying Chebyshev inequality to X, and taking into account (6) and (8), we have $\forall\epsilon > 0$

$$\Pr\left(\left|\mathbf{w}^\top\mathbf{Q}\mathbf{w} - \mathrm{Tr}(\mathbf{Q})\sigma^2\right| > t\sqrt{2d}\sigma^2\right) \leq \frac{\left(d + \mathrm{Tr}(\mathbf{Q}^2)\right)\sigma^4}{t^2\cdot 2d\sigma^4}.$$

Since $\mathbf{Q}^2$ is orthogonal, its trace does not exceed d, and we obtain (5). □

Remark 2.1. When $\sigma^2 = \frac{1}{d}$, the inequality (5) becomes:

$$\Pr\left(\left|\mathbf{w}^\top\mathbf{Q}\mathbf{w} - \frac{\mathrm{Tr}(\mathbf{Q})}{d}\right| > t\sqrt{\frac{2}{d}}\right) \leq \frac{1}{t^2}. \tag{10}$$

Corollary 1.1. *Let* $\mathbf{w} \sim \mathcal{N}(0, \frac{1}{d}\mathbf{I})$*, then* $\forall t > 0$

$$\Pr\left(\left|\|\mathbf{w}\|^2 - 1\right| > t\sqrt{\frac{2}{d}}\right) \leq \frac{1}{t^2}. \tag{11}$$

Proof. The statement follows from Lemma 1 and its Remark 2.1, when $\mathbf{Q} = \mathbf{I}$. □

Now we are ready to show that the partition function in (3) concentrates around $1 + \frac{1}{2d}$.

Lemma 2. *Let* Z_j *be a partition function in* (3)*, i.e.* $Z_j = \sum_{i=1}^n p_i e^{\mathbf{w}_j^\top\mathbf{c}_i}$*. Then*

$$Z_j \approx 1 + \frac{1}{2d} \quad (\forall j). \tag{12}$$

Proof. We will first show that the conditional expectation $\mathbb{E}[Z_j \mid \mathbf{w}_j]$ depends on $\mathbf{w}_j$ mainly through its norm $\|\mathbf{w}_j\|$:

$$\begin{aligned}\mathbb{E}\left[Z_j \mid \mathbf{w}_j\right] &= \mathbb{E}\left[\sum_{i=1}^{n} p_i e^{\mathbf{w}_j^\top \mathbf{c}_i} \;\middle|\; \mathbf{w}_j\right] = p_j\,\mathbb{E}\left[e^{\mathbf{w}_j^\top \mathbf{c}_j} \mid \mathbf{w}_j\right] + \sum_{i\neq j} p_i\,\mathbb{E}\left[e^{\mathbf{w}_j^\top \mathbf{c}_i} \mid \mathbf{w}_j\right] \\ &= p_j\,\mathbb{E}\left[e^{\mathbf{w}_j^\top \mathbf{Q}\mathbf{w}_j} \mid \mathbf{w}_j\right] + \sum_{i\neq j} p_i M_{\mathbf{c}_i}(\mathbf{w}_j) = p_j e^{\mathbf{w}_j^\top \mathbf{Q}\mathbf{w}_j} + \sum_{i\neq j} p_i e^{\frac{1}{2}\mathbf{w}_j^\top (\frac{1}{d}\mathbf{I})\mathbf{w}_j} \\ &= p_j e^{\mathbf{w}_j^\top \mathbf{Q}\mathbf{w}_j} + e^{\frac{1}{2d}\|\mathbf{w}_j\|^2}\sum_{i\neq j} p_i = p_j\left(e^{\mathbf{w}_j^\top \mathbf{Q}\mathbf{w}_j} - e^{\frac{1}{2d}\|\mathbf{w}_j\|^2}\right) + e^{\frac{1}{2d}\|\mathbf{w}_j\|^2}, \qquad (13)\end{aligned}$$

where $M_{\mathbf{c}_i}(\mathbf{w}_j)$ is the moment-generating function of $\mathbf{c}_i$ at $\mathbf{w}_j$. In Lemma 1 and Corollary 1.1 it is shown that $\mathbf{w}_j^\top \mathbf{Q}\mathbf{w}_j$ and $\|\mathbf{w}_j\|^2$ concentrate well around their means $\mathrm{Tr}(\mathbf{Q})\frac{1}{d}$ and 1 respectively and thus we can approximate

$$\mathbf{w}_j^\top \mathbf{Q}\mathbf{w}_j \approx \frac{\mathrm{Tr}(\mathbf{Q})}{d}, \qquad (14)$$

$$\|\mathbf{w}_j\|^2 \approx 1. \qquad (15)$$

The quantity $\frac{1}{2d}$ is small for $d \geq 50$ which is typical for the dimensionality of word vectors (Mikolov et al., 2013). Thus, using (14), (15), and Maclaurin expansion for $x \mapsto e^x$ in the last term of (13), we obtain

$$\mathbb{E}\left[Z_j \mid \mathbf{w}_j\right] \approx p_j\left(e^{\frac{\mathrm{Tr}(\mathbf{Q})}{d}} - e^{\frac{1}{2d}}\right) + 1 + \frac{1}{2d}. \qquad (16)$$

This approximation is very helpful as the right-hand side does not contain $\mathbf{w}_j$ and thus it is an approximation for $\mathbb{E}[Z_j]$ as well. Let $H_{n,\alpha}$ be the normalizer in (4), then

$$H_{n,\alpha} = \sum_{k=1}^{n} \frac{1}{k^{1-\alpha}} \sim \int_1^n \frac{dx}{x^{1-\alpha}} \sim \frac{n^\alpha}{\alpha},$$

and thus we have[3]

$$p_j \sim \frac{\alpha}{r_j^{1-\alpha} n^\alpha} \to 0 \text{ as } n \to \infty. \qquad (17)$$

Now, combining (16) and (17), we get

$$\mathbb{E}\left[Z_j \mid \mathbf{w}_j\right] \approx 1 + \frac{1}{2d}. \qquad (18)$$

Now let us show that $\mathrm{Var}[Z_j]$ is small relative to the mean $\mathbb{E}[Z_j]$. First, we have

$$\mathrm{Var}\left[Z_j \mid \mathbf{w}_j\right] = \sum_{i=1}^{n} p_i^2\,\mathrm{Var}\left[e^{\mathbf{w}_j^\top \mathbf{c}_i} \;\middle|\; \mathbf{w}_j\right] + \sum_{i\neq k} p_i p_k\,\mathrm{Cov}\left[e^{\mathbf{w}_j^\top \mathbf{c}_i},\, e^{\mathbf{w}_j^\top \mathbf{c}_k} \;\middle|\; \mathbf{w}_j\right]. \qquad (19)$$

3. We abuse the notation here and use '$\sim$' to denote asymptotic equivalence, i.e. $f(n) \sim g(n)$ iff $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$.

For the variance terms we have

$$\operatorname{Var}\left[e^{\mathbf{w}_j^\top \mathbf{c}_j} \mid \mathbf{w}_j\right] = \operatorname{Var}\left[e^{\mathbf{w}_j^\top \mathbf{Q}\mathbf{w}_j} \mid \mathbf{w}_j\right] = 0, \tag{20}$$

and

$$\begin{aligned}\operatorname{Var}\left[e^{\mathbf{w}_j^\top \mathbf{c}_i} \mid \mathbf{w}_j\right] &= \mathbb{E}\left[e^{2\mathbf{w}_j^\top \mathbf{c}_i} \mid \mathbf{w}_j\right] - \left(\mathbb{E}\left[e^{\mathbf{w}_j^\top \mathbf{c}_i} \mid \mathbf{w}_j\right]\right)^2 \\ &= M_{\mathbf{c}_i}(2\mathbf{w}_j) - \left(M_{\mathbf{c}_i}(\mathbf{w}_j)\right)^2 \\ &= e^{\frac{2}{d}\|\mathbf{w}_j\|^2} - e^{\frac{1}{d}\|\mathbf{w}_j\|^2}. \end{aligned} \tag{21}$$

Conditioned on $\mathbf{w}_j$, the random variables $\{e^{\mathbf{w}_j^\top \mathbf{c}_i}\}_{i \neq j}$ are independent, while $e^{\mathbf{w}_j^\top \mathbf{c}_j}$ is constant, and thus

$$\operatorname{Cov}\left[e^{\mathbf{w}_j^\top \mathbf{c}_i}, e^{\mathbf{w}_j^\top \mathbf{c}_k} \mid \mathbf{w}_j\right] = 0, \quad i \neq k. \tag{22}$$

From (19), (20), (21), and (22), we have

$$\operatorname{Var}[Z_j \mid \mathbf{w}_j] = \left(e^{\frac{2}{d}\|\mathbf{w}_j\|^2} - e^{\frac{1}{d}\|\mathbf{w}_j\|^2}\right) \sum_{i \neq j} p_i^2. \tag{23}$$

Notice that for some constant $C > 0$

$$\sum_{i \neq j} p_i^2 \sim \frac{C}{n^{2\alpha}} \int_1^n \frac{dx}{x^{2-2\alpha}} \sim \frac{C}{n^{\min\{1,\, 2\alpha\}}}. \tag{24}$$

Combining (15), (23), (24), and using Maclaurin expansion for $x \mapsto e^x$, we have

$$\operatorname{Var}[Z_j \mid \mathbf{w}_j] \sim \frac{C}{dn^{\min\{1,\, 2\alpha\}}} \to 0 \quad \text{as} \quad n \to \infty. \tag{25}$$

Now, the statement of the lemma follows from (18) and (25). □

Remark 2.2. Lemma 2 basically says that under Assumptions 1 and 2, the model (3) self-normalizes, i.e. the normalization term is almost constant and, moreover, it is almost 1. This result is similar to the result of Andreas and Klein (2015), but differs in that our model (3) is not log-linear as its condition (j) and prediction (i) are both parameterized. The result of Goldberger and Melamud (2018) on self-normalization of the NCE language models is closer to ours but the setup differs in that p_i does not appear as a factor in their model. We finally notice that Lemma 2 is an analogue of Lemma 2.1 from Arora et al. (2016) but adapted to our settings.

2.2 Q is an Involutary Matrix

Lemma 3. *Let* $\mathbf{Q}$ *be the matrix mapping word vectors to context vectors as in* (2)*. Then, under Assumptions 1, 2, and 3,* $\mathbf{Q}$ *is approximately an involutary matrix.*

Proof. The dimensionality d of word vectors is usually in the range $[50, 1000]$ (Mikolov et al., 2013), and thus we can neglect the term $\frac{1}{2d}$ in (12) and approximate $Z_j \approx 1$. This means that the model (3) simplifies to

$$p(i \mid j) \approx p(i) \cdot e^{\mathbf{w}_j^\top \mathbf{c}_i},$$

or, equivalently

$$\ln \frac{p(i,j)}{p(i)p(j)} \approx \mathbf{c}_i^\top \mathbf{w}_j. \tag{26}$$

where $p(i,\, j)$ is the probability that the words i and j co-occur in the same context window. Notice that the left-hand side in (26) is the pointwise mutual information (PMI) between words i and j. From (2) and (26) we have

$$\mathrm{PMI}(i,j) \approx \mathbf{w}_i^\top \mathbf{Q}^\top \mathbf{w}_j \quad \Leftrightarrow \quad \mathrm{PMI} \approx \mathbf{W}\mathbf{Q}^\top \mathbf{W}^\top,$$

where PMI stands for the PMI-matrix, and $\mathbf{W}$ is a $n \times d$ matrix in which i-th row is $\mathbf{w}_i^\top$. Since $p(i,j) = p(j,i)$, we should have $\mathrm{PMI} = \mathrm{PMI}^\top$, which implies

$$\begin{aligned} \mathbf{W}\mathbf{Q}^\top \mathbf{W}^\top &\approx \mathbf{W}\mathbf{Q}\mathbf{W}^\top \\ \Leftrightarrow \mathbf{W}^\top \mathbf{W}\mathbf{Q}^\top \mathbf{W}^\top \mathbf{W} &\approx \mathbf{W}^\top \mathbf{W}\mathbf{Q}\mathbf{W}^\top \mathbf{W} \\ \Leftrightarrow \mathbf{Q}^\top &\approx \mathbf{Q}, \end{aligned} \tag{27}$$

where we used the fact that $\mathbf{W}^\top \mathbf{W} \approx \mathbf{I}$. Since $\mathbf{Q}$ is assumed to be orthogonal, from (27) we get

$$\mathbf{Q}^2 \approx \mathbf{I}.$$

Thus, $\mathbf{Q}$ is approximately an involutary matrix, and we can choose it to be a signature matrix, i.e. a diagonal matrix with ± 1 on the diagonal[4]:

$$\mathbf{Q} := \mathrm{diag}(\pm 1, \ldots, \pm 1). \tag{28}$$

□

In this way, context vectors are word vectors with some of the coordinates being multiplied by -1. The natural question is "How many of the coordinates should be flipped?"

2.3 PMI as a Random Matrix

Let $\mathbf{x}_i \in \mathbb{R}^l$ be a vector consisting of the first l coordinates of $\mathbf{w}_i$, i.e.

$$\mathbf{x}_i = \mathbf{w}_{i,1:l} = \left[w_{i1}, \ldots, w_{il}\right], \tag{29}$$

and let $\mathbf{y}_i \in \mathbb{R}^{d-l}$ be a vector consisting of the last $d-l$ coordinates of $\mathbf{w}_i$, i.e.

$$\mathbf{y}_i = \mathbf{w}_{i,\, l+1:d} = \left[w_{i(l+1)}, \ldots, w_{id}\right]. \tag{30}$$

Due to Assumption 1, $\mathbf{x}_i$'s are i.i.d. draws from $\mathcal{N}\left(0, \frac{1}{d}\mathbf{I}_{l\times l}\right)$, $\mathbf{y}_i$'s are i.i.d. draws from $\mathcal{N}\left(0, \frac{1}{d}\mathbf{I}_{(d-l)\times(d-l)}\right)$, and $\{\mathbf{x}_i\}$, $\{\mathbf{y}_i\}$ are jointly independent. Without restricting the generality, assume that the first l diagonal elements in (28) are equal to $+1$, and the rest $d-l$ elements are equal to -1. Thus

$$\mathbf{w}_i^\top \mathbf{Q}^\top \mathbf{w}_j = \mathbf{x}_i^\top \mathbf{x}_j - \mathbf{y}_i^\top \mathbf{y}_j.$$

4. One can show that any involutary matrix can be represented as $\mathbf{P}^\top \mathrm{diag}(\pm 1, \ldots, \pm 1)\mathbf{P}$, where $\mathbf{P}$ is orthogonal, and thus by reparametrization $\widetilde{\mathbf{w}}_i = \mathbf{P}\mathbf{w}_i \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \frac{1}{d}\mathbf{I})$, we can still have (28).

For $i \neq j$, $\mathbf{x}_i^\top \mathbf{x}_j$ is a sum of l i.i.d. random variables with the mean 0 and the variance $\frac{1}{d^2}$, and by Central Limit Theorem, $\mathbf{x}_i^\top \mathbf{x}_j \approx \mathcal{N}\left(0, \frac{l}{d^2}\right)$. Similarly, $\mathbf{y}_i^\top \mathbf{y}_j \approx \mathcal{N}\left(0, \frac{d-l}{d^2}\right)$, and thus

$$\mathbf{w}_i^\top \mathbf{Q}^\top \mathbf{w}_j \approx \mathcal{N}\left(0, \frac{1}{d}\right), \quad i \neq j. \tag{31}$$

For $i = j$, we have

$$\mathbf{w}_i^\top \mathbf{Q}^\top \mathbf{w}_i = \|\mathbf{x}_i\|^2 - \|\mathbf{y}_i\|^2 \sim \frac{1}{d}(\chi_l^2 - \chi_{d-l}^2) \approx \mathcal{N}\left(\frac{2l-d}{d}, \frac{2}{d}\right), \tag{32}$$

where χ_l^2 is a chi-square random variable with l degrees of freedom. By combinatorial argument (similar to that of Lemma 1) one can show that covariance between any two distinct and non-symmetric entries of $\mathbf{W}\mathbf{Q}^\top\mathbf{W}^\top$ is zero, and thus

$$\begin{aligned} &\mathrm{Cov}\left[\mathrm{PMI}_{ij}, \mathrm{PMI}_{pq}\right] \approx 0 \\ &\forall\, (i,j) \neq (p,q),\, (p,q) \neq (j,i) \end{aligned} \tag{33}$$

Moreover, we can show that PMI_{ij} and PMI_{pq} tend to be independent when d is large enough. For the case $i \neq p$, $j \neq q$ this follows directly from Assumption 1. Now consider the case $i = p$, $j \neq q$ (two distinct elements from the same row). The case $i \neq p$, $j = q$ (two distinct elements from the same column) can be analyzed similarly. Let $\mathbf{t} = \begin{bmatrix} t_1 & t_2 \end{bmatrix}^\top$. Then the moment-generating function (m.g.f.) of $\begin{bmatrix} \mathrm{PMI}_{ij} & \mathrm{PMI}_{iq} \end{bmatrix}^\top$ at $\mathbf{t}$ is

$$\begin{aligned} M_{[\mathrm{PMI}_{ij}\ \mathrm{PMI}_{iq}]^\top}(\mathbf{t}) &= \mathbb{E}\left[e^{t_1 \mathbf{w}_i^\top \mathbf{c}_j + t_2 \mathbf{w}_i^\top \mathbf{c}_q}\right] = \mathbb{E}\left[\mathbb{E}\left[e^{t_1 \mathbf{w}_i^\top \mathbf{c}_j + t_2 \mathbf{w}_i^\top \mathbf{c}_j} \mid \mathbf{w}_i\right]\right] \\ &= \mathbb{E}\left[M_{\mathbf{c}_j}(t_1 \mathbf{w}_i) \cdot M_{\mathbf{c}_q}(t_2 \mathbf{w}_i)\right] = \mathbb{E}\left[e^{\frac{1}{2d} t_1^2 \|\mathbf{w}_i\|^2} \cdot e^{\frac{1}{2d} t_2^2 \|\mathbf{w}_i\|^2}\right] \\ &= \mathbb{E}\left[e^{\frac{1}{2d}\|\mathbf{t}\|^2 \|\mathbf{w}_i\|^2}\right] = M_{\chi_d^2}\left(\frac{1}{2d^2}\|\mathbf{t}\|^2\right) = \left(1 - \frac{\|\mathbf{t}\|^2}{d^2}\right)^{-\frac{d}{2}} \\ &= e^{\frac{\|\mathbf{t}\|^2}{2d}}\left[1 + O\left(\frac{1}{d^3}\right)\right], \end{aligned} \tag{34}$$

and the last expression for large d is approximately $e^{\frac{\|\mathbf{t}\|^2}{2d}}$ which is the m.g.f. of a two-dimensional Gaussian vector with the distribution $\mathcal{N}(\mathbf{0}, \frac{1}{d}\mathbf{I}_{2\times 2})$. Hence

$$\begin{bmatrix} \mathrm{PMI}_{ij} & \mathrm{PMI}_{iq} \end{bmatrix}^\top \approx \mathcal{N}\left(\mathbf{0}, \tfrac{1}{d}\mathbf{I}_{2\times 2}\right), \tag{35}$$

which implies approximate independence between PMI_{ij} and PMI_{iq}. Hence, from (31) and (32) we conclude that for the PMI matrix

- the above-diagonal entries have (approximately) the distribution $\mathcal{N}\left(0, \frac{1}{d}\right)$,
- the diagonal entries have (approximately) the distribution $\mathcal{N}\left(\frac{2l-d}{d}, \frac{2}{d}\right)$,
- all entries on and above its diagonal tend to pairwise independence.

This means that the PMI matrix is an approximately symmetric Gaussian random matrix with weakly dependent entries and it is known that the empirical distribution of eigenvalues of such matrix approaches a *symmetric around* 0 distribution as its size n increases (de Monvel & Khorunzhy, 1999).[5] Thus, we should have

$$\mathrm{Tr}(\mathrm{PMI}) \approx 0 \quad \Leftrightarrow \quad \sum_{i=1}^{n} \mathbf{w}_i^\top \mathbf{Q}^\top \mathbf{w}_i \approx 0 \quad \Leftrightarrow \quad \sum_{i=1}^{n} \|\mathbf{x}_i\|^2 \approx \sum_{i=1}^{n} \|\mathbf{y}\|^2. \tag{36}$$

Recall, that $d\|\mathbf{x}_i\|^2 \sim \chi^2_l$ and $d\|\mathbf{y}_i\|^2 \sim \chi^2_{d-l}$. Hence, taking expectation on both sides of (36) we have

$$\frac{n}{d}\,\mathbb{E}\left[\chi^2_l\right] \approx \frac{n}{d}\,\mathbb{E}\left[\chi^2_{d-l}\right] \quad \Leftrightarrow \quad l \approx d - l \quad \Leftrightarrow \quad l \approx \frac{d}{2}. \tag{37}$$

which concludes the proof of Theorem 1.

Remark 2.3. In terms of the introduced notation (29) and (30), each word's vector $\mathbf{w}_i$ splits into two subvectors $\mathbf{x}_i$ and $\mathbf{y}_i$, and due to Theorem 1, our model (3) for generating a word i in the context of a word j can be rewritten as

$$p(i \mid j) \approx p_i \cdot e^{\mathbf{x}_j^\top \mathbf{x}_i - \mathbf{y}_j^\top \mathbf{y}_i}.$$

Interestingly, embeddings of the first type ($\mathbf{x}_i$ and $\mathbf{x}_j$) are responsible for pulling the word i into the context of the word j, while embeddings of the second type ($\mathbf{y}_i$ and $\mathbf{y}_j$) are responsible for pushing the word i away from the context of the word j.

3. Empirical Verification

In this section we empirically verify two predictions of our theory: a symmetric distribution of a PMI spectrum and the involutarity of a matrix transforming input embeddings into output ones.

3.1 Symmetry of a PMI Spectrum

To verify that the real-world PMI matrices have indeed a symmetric (around 0) distribution of their eigenvalues, we consider two widely-used datasets, `text8` and `enwik9`,[6] from which we extract PMI matrices using the HYPERWORDS tool of Levy et al. (2015). We use the default settings for all hyperparameters, except the word frequency threshold and context window size. We were ignoring words that appeared less than 100 times and 150 times in `text8` and `enwik9` correspondingly, resulting in vocabularies of 11,815 and 29,145 correspondingly. We additionally experiment with the context window size 5, which by default is set to 2, and which we believe could affect the results. By default, HYPERWORDS zeroes out an entry $\mathrm{PMI}_{i,j}$ if the words i and j do not co-occur in the corpus.[7] The eigenvalues of

5. Usually papers on random matrix theory do not state this explicitly. Rather they show that the limiting distribution has odd moments equal to 0. This is the case for the referenced paper as well, see the proof of their Theorem 2.2.
6. `http://mattmahoney.net/dc/textdata.html`. The `enwik9` data was processed with the Perl-script WIKIFIL.PL provided on the same webpage. It filters Wikipedia XML dumps to a "clean" text consisting only of lowercase letters and spaces (never consecutive).
7. This means that apriori each i and j are assumed to be independent, and we follow this convention. Thus, our PMI matrices do not have any $-\infty$'s, instead, they have lots of 0's, i.e. they are sparse.

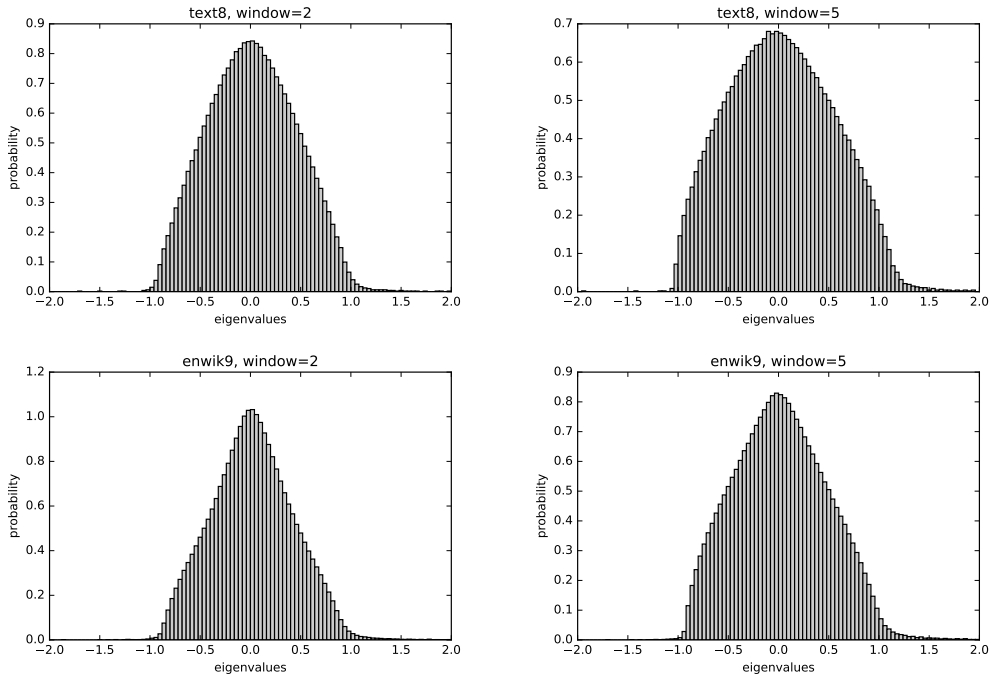

Figure 2: Empirical distribution of eigenvalues of PMI matrices.

Data set	Size	T	$\lvert\mathcal{W}\rvert$
text8	100 MB	17M	254K
enwik9	715 MB	124M	833K

Table 1: Corpus statistics. T = total length in tokens; $|\mathcal{W}|$ = number of unique words.

the PMI matrices are calculated using the TensorFlow library, and the above-mentioned threshold of 150 for `enwik9` was chosen to fit the resulting PMI matrix into the GPU memory (12GB, NVIDIA Titan X Maxwell). The histograms of eigenvalues are provided in Figure 2. As we can see, the distributions are not perfectly symmetric with a little right skewness, but in general they seem to be symmetric. Notice, that this is in stark contrast with the equation (2.5) from Arora et al. (2016), which claims that the PMI matrix should be approximately positive semi-definite, i.e. that it should have mostly positive eigenvalues. Also, notice that the shapes of distributions are far from resembling the Wigner semicircle law $x \mapsto \frac{1}{2\pi}\sqrt{4-x^2}$, which is the limiting distribution for the eigenvalues of many random symmetric matrices with i.i.d. entries (Wigner, 1955, 1958). This means that the entries of a typical PMI matrix *are* dependent, otherwise we would observe approximately semicircle distributions for its eigenvalues. Interestingly, there is a striking similarity between the shapes of distributions in Figure 2 and of spectral densities of the scale-free random graphs with strong clustering (Farkas, Derényi, Barabási, & Vicsek, 2001) which arise in physics and network science. Notice that the connection between human language structure and

scale-free random graphs with strong clustering was observed previously by Cancho and Solé (2001), and we believe it is worth investigating this connection deeper.

3.2 Involutarity of Q for SGNS Embeddings

Assumption 2 together with Lemma 3 suggest that the matrix $\mathbf{Q}$ should be approximately involutary. To test this claim we train off-the-shelf skip-gram embeddings $\{\mathbf{w}_i\}$ and $\{\mathbf{c}_i\}$ on `text8` and `enwik9` datasets using the reference WORD2VEC implementation from the TENSORFLOW codebase[8]. Let $\mathbf{W} \in \mathbb{R}^{n\times d}$ be a word embedding matrix in which i-th row is $\mathbf{w}_i^\top$, and $\mathbf{C} \in \mathbb{R}^{n\times d}$ be a context embedding matrix in which i-th row is $\mathbf{c}_i^\top$. According to (2),

$$\mathbf{C}^\top = \mathbf{Q}\mathbf{W}^\top \quad \Leftrightarrow \quad \mathbf{C} = \mathbf{W}\mathbf{Q}^\top.$$

Thus, $\hat{\mathbf{Q}} := \mathbf{W}^\dagger\mathbf{C}$ should give an approximately involutary matrix, where $\mathbf{W}^\dagger$ is the pseudo-inverse of $\mathbf{W}$. This means that $\hat{\mathbf{Q}}^2$ should be approximately an identity matrix. The distribution of diagonal and off-diagonal elements of $\hat{\mathbf{Q}}^2$ is given in Fig. 3. We see that the

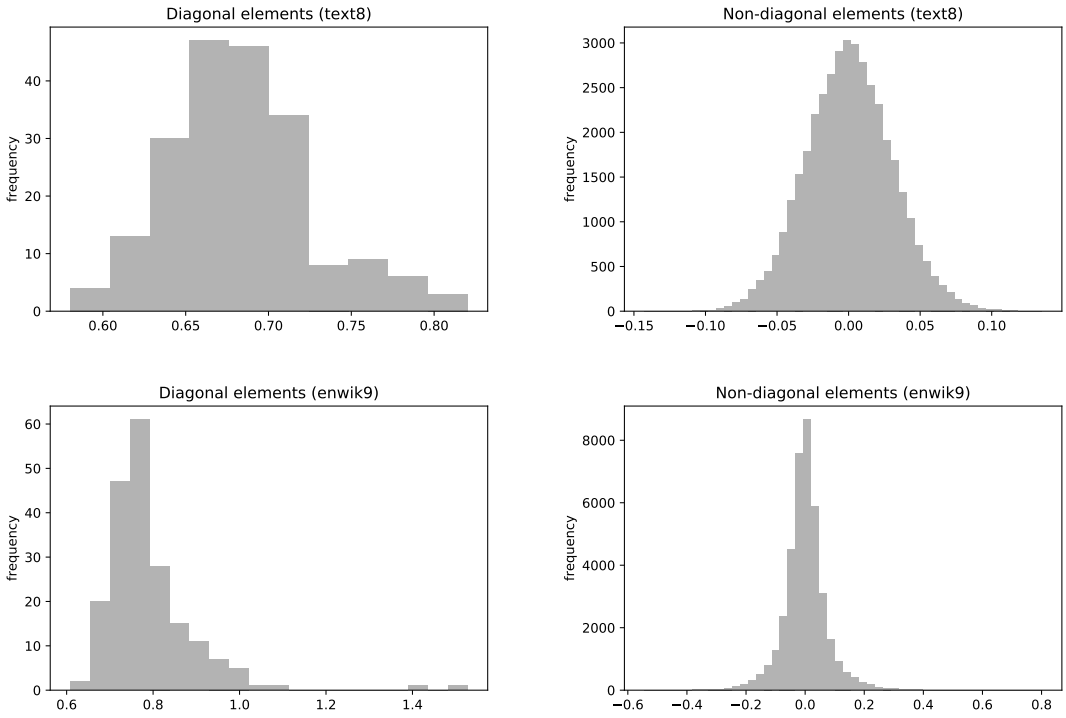

Figure 3: Diagonal and off-diagonal elements of $(\mathbf{W}^\dagger\mathbf{C})^2$.

diagonal elements are concentrated around their means 0.68 (`text8`) and 0.79 (`enwik9`),

8. `https://github.com/tensorflow/models/blob/master/tutorials/embedding`

while the off-diagonal elements are concentrated around 0, i.e.

$$\hat{\mathbf{Q}}^2 \approx \begin{cases} 0.68 \cdot \mathbf{I} & \text{for } \texttt{text8} \\ 0.79 \cdot \mathbf{I} & \text{for } \texttt{enwik9} \end{cases}.$$

However, our theory predicts $\hat{\mathbf{Q}}^2 \approx \mathbf{I}$. We attribute this mismatch to the underlying gap between Assumption 2 and the empirical observations: in SGNS the transform between word and context vectors is *not exactly* orthogonal, since

- the pairwise similarities between word and context embeddings are highly correlated (e.g. $\rho = 0.77$ on `text8`), but are not exactly equal: $\cos\angle(\mathbf{w}_i, \mathbf{w}_j) \neq \cos\angle(\mathbf{c}_i, \mathbf{c}_j)$;
- norms of word vectors and norms of context vectors are highly correlated ($\rho = 0.76$ on `text8`), but are not exactly equal: $\|\mathbf{w}_i\| \neq \|\mathbf{c}_i\|$.

We stress here that our assumptions are *motivated by* but are not exactly consistent with the skip-gram embeddings. Despite this, our theory is quite applicable to the SGNS model as is shown in Section 4.

4. Weight Tying in the Skip-gram Model

Data	Model	Size	Finkelstein et al. WordSim	Bruni et al. MEN	Radinsky et al. M. Turk	Luong et al. Rare Words	Google	MSR
`text8`	SGNS	28M	.681	.241	.631	.072	.307	.286
	SGNS+WT	14M	.638	.216	.642	.058	.309	.319
	SGNS+WTR	14M	.637	.215	.624	.057	.314	.319
`enwik9`	SGNS	87M	.671	.268	.662	.213	.558	.410
	SGNS+WT	44M	.640	.237	.615	.188	.515	.425
	SGNS+WTR	44M	.633	.236	.639	.175	.516	.429

Table 2: Evaluation of word embeddings on the analogy tasks (Google and MSR) and on the similarity tasks (the rest). For word similarities evaluation metric is the Spearman's correlation with the human ratings, while for word analogies it is the percentage of correct answers. Model sizes are in number of trainable parameters.

We would like to apply our results to tie embeddings in the skip-gram model of Mikolov et al. (2013) in a theoretically grounded way. One may argue that our key Assumption 3 differs from the softmax-prediction of the skip-gram model. Although this is true, in fact the softmax normalization is never used in practice when training skip-gram. Instead it is common to replace the softmax cross-entropy by the negative sampling objective (equation (4) in Mikolov et al., 2013), and its optimization is almost equivalent to finding a low-rank approximation of the shifted word-word PMI matrix in the form $\mathbf{w}_i^\top \mathbf{c}_j \approx \mathrm{PMI}_{ij} - \log k$ (Levy & Goldberg, 2014b). Since our Assumptions lead to the same conclusion up to a constant shift (26), we believe that Theorem 1 can be directly applied to tie word ($\mathbf{w}_i$) and

context ($\mathbf{c}_i$) embeddings in the SGNS model. For this purpose we form a vector

$$\mathbf{q} = [\underbrace{+1, \dots, +1}_{d/2}, \underbrace{-1 \dots, -1}_{d/2}] \in \mathbb{R}^d$$

and then put

$$\mathbf{c}_i = \mathbf{q} \odot \mathbf{w}_i \tag{38}$$

for all words i in the vocabulary. This is equivalent to (2) when the matrix $\mathbf{Q}$ has a special diagonal form (28) with the first $d/2$ diagonal entries being $+1$ and the rest $d/2$ entries being -1. Such modification of the SGNS is refered to as 'SGNS + WT'. We also experiment with random flipping of signs: for this purpose we form $\mathbf{q} \in \mathbb{R}^d$ as a random vector consisting of d i.i.d. draws from the Rademacher distribution[9] and then put $\mathbf{c}_i$ as in (38). Such variant of weight tying is referred to as 'SGNS + WTR'.

The word embeddings $\mathbf{w}_i$ are initialized randomly, and then trained on `text8` and `enwik9` using the reference WORD2VEC implementation from the TENSORFLOW codebase[10] with all hyperparameters set to their default values[11] except that we choose the learning rate to decay 20% faster in the weight-tied model. This additional tuning of the learning rate decay is not surprising: the model with tied embeddings has two times fewer parameters compared to the model with untied weights, and this leads to a significant change of the optimization landscape, which in turn results in the need to tune the most sensitive hyperparameter — the learning rate (or its decay schedule). As is standard nowadays the trained embeddings are evaluated on several word similarity and word analogy tasks. We used the HYPERWORDS tool of Levy et al. (2015) and we refer the reader to their paper for the methodology of evaluation. We only mention here a few key points:

- Our goal is not to beat state of the art, but to empirically validate the statement of Theorem 1. This is why we were evaluating only word (input) embeddings for both SGNS and SGNS+WT. I.e., we were not adding context vectors to word vectors in the similarity tasks, as it is usually done nowadays.
- Word similarity datasets contain word pairs together with human-assigned similarity scores. The word vectors are evaluated by ranking the pairs according to their cosine similarities and measuring the correlation (Spearman's ρ) with the human ratings.
- For answering analogy questions (a is to b as c is to ?) we use the 3COSMUL of Levy and Goldberg (2014a) and the evaluation metric for the analogy questions is the percentage of correct answers.

The results of evaluation are provided in Table 2. First of all, notice that random flipping of signs (SGNS+WTR) gives practically the same results as non-random flipping (SGNS+WT). Next, SGNS+WT produces embeddings comparable in quality with those produced by the baseline SGNS model despite having 50% fewer parameters. This also

9. Rademacher distribution is a discrete probability distribution where a random variate X has a 50% chance of being $+1$ and a 50% chance of being -1.
10. `https://github.com/tensorflow/models/blob/master/tutorials/embedding/word2vec.py`
11. Embedding size $d = 200$, 15 epochs to train, initial learning rate 0.2 on `text8` and 0.15 on `enwik9`, 100 negative samples per training example, batch size 16, windows size 5.

empirically validates the statement of our Theorem 1. We notice that similar results can be obtained by letting the linear transform $\mathbf{Q}$ be a trainable matrix as shown by Gulordava et al. (2018). The main difference of our approach is that we know exactly the form of $\mathbf{Q}$, and thus we do not need to learn it.

Remark 4.1 (**Proportion of flipped coordinates**). We study how changing the proportion of $+1$'s and -1's affects the quality of word vectors. For this purpose we vary the percentage of $+1$'s from 0% to 100% in $\mathbf{q} \in \mathbb{R}^{200}$, with the rest of coordinates being -1's and evaluate the resulting vectors. The results are provided on Fig. 4. Interestingly, there is

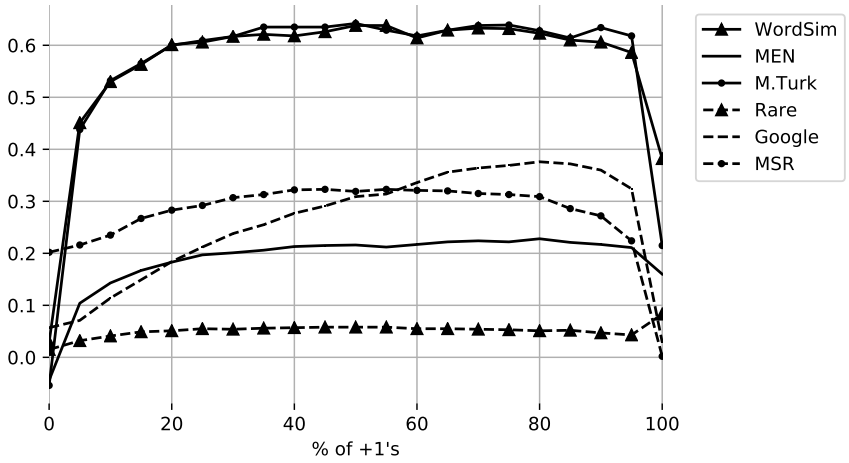

Figure 4: Proportion of non-flipped coordinates vs performance on tasks.

no significant change in the overall quality of word embeddings as long as we have 35%–65% of $+1$'s in $\mathbf{q}$, while the quality deteriorates drastically when there are $< 25\%$ or $> 90\%$ of $+1$'s.

Remark 4.2 (**Factorizing PMI vs training SGNS**). Instead of training SGNS from scratch with tied weights, we can factorize the PMI matrix to get $\mathbf{W}$ and $\mathbf{Q}$ directly: since the PMI matrix is symmetric, its eigendecomposition has the following form:

$$\mathrm{PMI} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top,$$

where $\mathbf{U} \in \mathbb{R}^{n \times n}$ is orthogonal, and $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$ is diagonal with eigenvalues of PMI on the diagonal. Assume that the eigenvalues are sorted in descending order of their absolute values, then the rank d approximation of PMI is

$$\mathrm{PMI}_d = \mathbf{U}_{1:n,1:d}\mathbf{\Lambda}_{1:d,1:d}\mathbf{U}^\top_{1:n,1:d}.$$

Notice that due to symmetry of PMI this approximation is equivalent to the truncated singular value decomposition (SVD) of PMI and, by the Eckart-Young theorem (Eckart & Young, 1936), PMI_d is the closest rank-d matrix to PMI in Frobenius norm, i.e. $\|\mathrm{PMI} - \mathrm{PMI}_d\|_F \leq \|\mathrm{PMI} - \mathbf{A}\|_F$, $\forall \mathbf{A} \in \mathbb{R}^{n \times n} : \mathrm{rank}(\mathbf{A}) = d$. Thus we can choose $\mathbf{W}$ and $\mathbf{Q}$ as follows:

$$\mathbf{W} = \mathbf{U}_{1:n,1:d}\sqrt{|\mathbf{\Lambda}_{1:d,1:d}|}, \qquad \mathbf{Q} = \operatorname{sgn}\mathbf{\Lambda}_{1:d,1:d},$$

where $\sqrt{\cdot}$, $|\cdot|$ and $\mathrm{sgn}(\cdot)$ are applied element-wise. This will give essentially the same embeddings as in the work of Levy and Goldberg (2014b) who already showed that truncated SVD of the PMI matrix produces word vectors comparable in quality with those from the SGNS. Hence, we do not pursue this direction further.

Remark 4.3 (**Weight Tying in LSTM Language Models**). One may argue that our focus on SGNS is outdated because with ELMo (Peters et al., 2018) and other contextualized word representations, static representations like WORD2VEC may seem obsolete. Since ELMo representations are based on training a variant of a recurrent neural network, we perform an experiment on tying input and output embeddings in an LSTM-based language model (Zaremba, Sutskever, & Vinyals, 2014) via an involutory matrix (LSTM+WTQ) and compare this to a common way of equalizing input and output embeddings (LSTM+WT). We use perplexity to evaluate the performance of the language models. Both models are trained and evaluated on the `PTB` (Marcus, Marcinkiewicz, & Santorini, 1993) and the `WikiText-2` (Merity, Xiong, Bradbury, & Socher, 2017) data sets. For the `PTB` we utilize the standard training, validation, and test splits along with pre-processing per Mikolov et al. (2010). `WikiText-2` is an alternative to `PTB`, which is approximately two times as large in size and three times as large in vocabulary. We use the reference PyTorch implementation[12] for the LSTM+WT and modify it using the i.i.d. draws from the Rademacher distribution (as in WORD2VEC) to get LSTM+WTQ. The results of evaluation are provided

Model	`PTB`		`WikiText-2`	
	$d = 200$	$d = 650$	$d = 200$	$d = 650$
LSTM + WT	86.1	76.2	99.6	86.6
LSTM + WTQ	86.5	76.4	100.4	86.7

Table 3: Evaluation of weight-tied LSTM language models.

in Table 3. As one can see, weight tying through the involutary matrix gives a slightly worse performance in language modeling than the conventional weight tying.

5. Conclusion

There is a remarkable relationship between human language and other branches of science, and we can get interesting and practical results by studying such relationships deeper. For example, the modern theory of random matrices is replete with theoretical results that can be immediately applied to models of natural language once such models are cast into the appropriate probabilistic setting, as is done in this paper.

Acknowledgements

The work of Zhenisbek Assylbekov has been funded by the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan, contract # 346/018-

12. `https://github.com/pytorch/examples/tree/master/word_language_model`

2018/33-28, IRN AP05133700. The authors would like to thank anonymous reviewers for their valuable feedback.

References

Allen, C., Balažević, I., & Hospedales, T. (2018). What the vec? towards probabilistically grounded embeddings. *arXiv preprint arXiv:1805.12164*.

Andreas, J., & Klein, D. (2015). When and why are log-linear models self-normalizing?. In *Proceedings of NAACL-HLT*, pp. 244–249.

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2016). A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, *4*, 385–399.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2001). A neural probabilistic language model..

Bruni, E., Boleda, G., Baroni, M., & Tran, N.-K. (2012). Distributional semantics in technicolor. In *Proceedings of ACL*, pp. 136–145. Association for Computational Linguistics.

Cancho, R. F. I., & Solé, R. V. (2001). The small world of human language. *Proceedings of the Royal Society B: Biological Sciences*, *268*(1482), 2261.

de Monvel, A. B., & Khorunzhy, A. (1999). On the norm and eigenvalue distribution of large random matrices. *The Annals of Probability*, *27*(2), 913–944.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211–218.

Farkas, I. J., Derényi, I., Barabási, A.-L., & Vicsek, T. (2001). Spectra of real-world graphs: Beyond the semicircle law. *Physical Review E*, *64*(2), 026704.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on information systems*, *20*(1), 116–131.

Gittens, A., Achlioptas, D., & Mahoney, M. W. (2017). Skip-gram-zipf+ uniform= vector additivity. In *Proceedings of ACL*, Vol. 1, pp. 69–76.

Goldberg, Y., & Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Goldberger, J., & Melamud, O. (2018). Self-normalization properties of language modeling. In *Proceedings of COLING*, pp. 764–773.

Gulordava, K., Aina, L., & Boleda, G. (2018). How to represent a word and predict it, too: Improving tied architectures for language modelling. In *Proceedings of EMNLP*, pp. 2936–2941.

Hashimoto, T. B., Alvarez-Melis, D., & Jaakkola, T. S. (2016). Word embeddings as metric recovery in semantic spaces. *Transactions of the Association for Computational Linguistics*, *4*, 273–286.

Inan, H., Khosravi, K., & Socher, R. (2017). Tying word vectors and word classifiers: A loss framework for language modeling. In *Proceedings of ICLR*.

Levy, O., & Goldberg, Y. (2014a). Linguistic regularities in sparse and explicit word representations. In *Proceedings of CoNLL*, pp. 171–180.

Levy, O., & Goldberg, Y. (2014b). Neural word embedding as implicit matrix factorization. In *Proceedings of NeurIPS*, pp. 2177–2185.

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, *3*, 211–225.

Luong, T., Socher, R., & Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of CoNLL*, pp. 104–113.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, *19*(2), 313–330.

Melamud, O., Dagan, I., & Goldberger, J. (2017). A simple language model based on pmi matrix approximations. In *Proceedings of EMNLP*, pp. 1860–1865.

Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2017). Pointer sentinel mixture models. In *Proceedings of ICLR*.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of INTERSPEECH*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS*, pp. 3111–3119.

Mimno, D., & Thompson, L. (2017). The strange geometry of skip-gram with negative sampling. In *Proceedings of EMNLP*, pp. 2873–2878.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pp. 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pp. 2227–2237.

Press, O., & Wolf, L. (2017). Using the output embedding to improve language models. In *Proceedings of EACL*, Vol. 2, pp. 157–163.

Radinsky, K., Agichtein, E., Gabrilovich, E., & Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pp. 337–346. ACM.

Wigner, E. P. (1955). Characteristic vectors of bordered matrices with infinite dimensions. *Annals of Mathematics*, 548–564.

Wigner, E. P. (1958). On the distribution of the roots of certain symmetric matrices. *Annals of Mathematics*, 325–327.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zipf, G. (1935). *The psycho-biology of language: an introduction to dynamic philology.* Houghton Mifflin.