A comparison of correspondence analysis with PMI-based word embedding methods

Qianqian Qi*

1. Department of Methodology and Statistics,
Faculty of Social Sciences, Utrecht University, Utrecht, The Netherlands
*Corresponding author
q.qi@uu.nl

David J. Hessen

1. Department of Methodology and Statistics,
Faculty of Social Sciences, Utrecht University, Utrecht, The Netherlands
d.j.hessen@uu.nl

Peter G. M. van der Heijden

1. Department of Methodology and Statistics,
Faculty of Social Sciences, Utrecht University, Utrecht, The Netherlands;
2. Southampton Statistical Sciences Research Institute,
University of Southampton, Highfield, Southampton, UK
p.g.m.vanderheijden@uu.nl

arXiv:2405.20895v1 [cs.CL] 31 May 2024

Abstract: Popular word embedding methods such as GloVe and Word2Vec are related to the factorization of the pointwise mutual information (PMI) matrix. In this paper, we link correspondence analysis (CA) to the factorization of the PMI matrix. CA is a dimensionality reduction method that uses singular value decomposition (SVD), and we show that CA is mathematically close to the weighted factorization of the PMI matrix. In addition, we present variants of CA that turn out to be successful in the factorization of the word-context matrix, i.e. CA applied to a matrix where the entries undergo a square-root transformation (ROOT-CA) and a root-root transformation (ROOTROOT-CA). An empirical comparison among CA- and PMI-based methods shows that overall results of ROOT-CA and ROOTROOT-CA are slightly better than those of the PMI-based methods.

Keywords: Word2Vec; GloVe; Variance stabilization; Overdispersion; Singular value decomposition

1 Introduction

Word embeddings, i.e., dense and low dimensional word representations, are useful in various natural language processing (NLP) tasks (Jurafsky & Martin, 2023; Sasaki, Heinzerling, Suzuki, & Inui, 2023). Three successful methods to derive such word representations are related to the factorization of the pointwise mutual information (PMI) matrix, an important matrix to be analyzed in NLP (Alqahtani, Al-Twairesh, & Alsanad, 2023; Bae et al., 2021; Egleston et al., 2021). The PMI matrix is a weighted version of the word-context co-occurrence matrix and measures how often two words, a target word and a context word, co-occur, compared with what we would expect if the two words were independent. The analysis of a positive PMI (PPMI) matrix, where all negative values in a PMI matrix are replaced with zero (Alqahtani et al., 2023; Turney & Pantel, 2010; Zhang, Palade, Wang, & Ji, 2022), generally leads to a better performance in semantic tasks (Bullinaria & Levy, 2007), and in most applications the PMI matrix is replaced with the PPMI matrix (Salle, Villavicencio, & Idiart, 2016).

The first method, PPMI-SVD, decomposes the PPMI matrix with a singular value decomposition (SVD) (Levy & Goldberg, 2014; Levy, Goldberg, & Dagan, 2015; Stratos, Collins, & Hsu, 2015; Zhang et al., 2022). The second one is GloVe (Pennington, Socher, & Manning, 2014). GloVe factorizes the logarithm of the word-context matrix with an adaptive gradient algorithm (AdaGrad) (Duchi, Hazan, & Singer, 2011). According to Shazeer, Doherty, Evans, and Waterson (2016); Shi and Liu (2014), GloVe is almost equivalent to factorizing a PMI matrix shifted by the logarithm of the sum of the elements of a word-context matrix. The third method is Word2Vec's skip-gram with negative sampling (SGNS) (Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). SGNS uses a neural network model to generate word embeddings. Levy and Goldberg (2014) proved that SGNS implicitly factorizes a PMI matrix shifted by the logarithm of the number of negative samples in SGNS.

In this paper we study what correspondence analysis (CA) (Beh & Lombardo, 2021; Greenacre, 2017) has to offer for the analysis of word-context co-occurrence matrices. CA is an exploratory statistical method that is often used for visualization of a low dimensional approximation of a matrix. It is close to the T-test weighting scheme (Curran, 2004; Curran & Moens, 2002), where standardized residuals are studied, as CA is based on the SVD of the matrix of standardized residuals. In the context of document-term matrices, CA has been compared earlier with latent semantic analysis (LSA), where the document-term matrix is also decomposed with an SVD (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Dumais, Furnas, Landauer, Deerwester, & Harshman, 1988). Although CA is similar to LSA, there is theoretical and empirical research showing that CA is to be preferred over LSA for text categorization and information retrieval (Qi, Hessen, Deoskar, & Van der Heijden, 2023; Qi, Hessen, & Van der Heijden, 2023).

CA of a two-way contingency table is equivalent to canonical correlation analysis (CCA) of the data in the form of indicator matrices for the row variable and the column variable of the two-way contingency table (Greenacre, 1984). Stratos et al. (2015) proposed to combine CCA with a square-root transformation of the cell frequencies of the contingency table. In this paper we refer to this procedure as ROOT-CCA, to distinguish it from ROOT-CA introduced later. Stratos et al. (2015) found that, on word similarity tasks, (1) the performance of CCA is quite bad, but the performance of ROOT-CCA is a marked improvement, and (2) ROOT-CCA outperforms PPMI-SVD, GloVe, and SGNS. However, CA has not yet been linked to PMI-based methods.

A document-term matrix has some similarity to a word-context matrix, as they both use counts. In this paper, mathematically, we show that CA is close to a weighted factorization of the PMI matrix. We also propose a direct weighted factorization of the PMI matrix (PMI-GSVD). Furthermore, we empirically compare the performance of CA with the performance of PMI-based methods on a word similarity task.

In the context of CA, Nishisato, Beh, Lombardo, and Clavel (2021) point out, generally speaking, a two-way contingency table is prone to overdispersion. Overdispersion may negatively affect the performance of CA (Beh, Lombardo, & Alberti, 2018; Nishisato et al., 2021). To deal with this overdispersion, a fourth-root transformation can be used (Field, Clarke, & Warwick, 1982; Greenacre, 2009, 2010). The fourth root transformation has been widely discussed and applied (Downing, 1981; France & Heung, 2023; Kostensalo et al., 2023). Therefore, in addition to the word-context matrix, CA is also applied to the fourth-root transformation of the word-context matrix (ROOTROOT-CA). Inspired by ROOT-CCA, CA is also applied to the square-root transformation of the word-context matrix (ROOT-CA). Recently, ROOT-CA has been explored in biology (Hsu & Culhane, 2023). The difference between ROOT-CCA and ROOT-CA is discussed in Section 3.3.

In the following section, research objectives are presented. In Section 3 CA, the three variants of CA, and the T-Test weighting scheme are introduced. The three PMI-based methods are described in Section 4. Theoretical relationships between CA and the PMI-based methods are shown in Section 5. In Section 6 we present two corpora to build word vectors and five word similarity datasets to evaluate word vectors. Section 7 illustrates the setup of the empirical study using these two corpora where CA, PMI-SVD, PPMI-SVD, PMI-GSVD, ROOT-CA, ROOTROOT-CA, ROOT-CCA, SGNS, and GloVe are compared. Section 8 presents the results for these methods on word similarity tasks. Section 9 concludes and discusses this paper.

2 Research objectives

Considering the foregoing, this study focuses on word embeddings in NLP. The objective is to explore the relationship between CA and PMI-based methods and compare the performance in word similarity tasks. In addition, we explore the performance of variants of CA, namely ROOT-CA and ROOTROOT-CA.

3 Correspondence analysis

In this section, first we describe correspondence analysis (CA) using a distance interpretation (Benzécri, 1973; Greenacre & Hastie, 1987), which is a popular way to present CA. Then we present CA making use of an objective function, thus making the later comparison with PMI-based methods straightforward. Third, we present three variants of CA in word embedding. Finally, the T-Test weighting scheme (Curran, 2004; Curran & Moens, 2002) is described, as it turns out to be remarkably similar to CA.

A word-context matrix is a matrix with counts, in which the rows and columns are labeled by terms. In each cell a count represents the number of times the row (target) word and the column (context) word co-occur in a text (Jurafsky & Martin, 2023). Consider a word-context matrix denoted as $\boldsymbol{X}$ having I rows ($i = 1, 2, \cdots, I$) and J columns ($j = 1, 2, \cdots, J$), where the element for row i and column j is x_{ij}. The joint observed proportion is $p_{ij} = x_{ij}/x_{++}$, where "+" represents the sum over the corresponding elements and $x_{++} = \sum_i \sum_j x_{ij}$. The marginal proportions of target word i and context word j are $p_{i+} = \sum_j p_{ij}$ and $p_{+j} = \sum_i p_{ij}$, respectively.

3.1 Introduction to CA

CA is an exploratory method for the analysis of two-way contingency tables. It allows to study how the counts in the contingency table depart from statistical independence. Here we introduce CA in the context of the word-context matrix $\boldsymbol{X}$. In CA of the matrix $\boldsymbol{X}$, first the elements x_{ij} are converted to joint observed proportions p_{ij}, and these are transformed into standardized residuals (Greenacre, 2017)

$$\frac{p_{ij} - p_{i+}p_{+j}}{\sqrt{p_{i+}p_{+j}}}. \tag{1}$$

Then an SVD is applied to this matrix of standardized residuals, yielding

$$\frac{p_{ij} - p_{i+}p_{+j}}{\sqrt{p_{i+}p_{+j}}} = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k u_{ik} v_{jk}, \tag{2}$$

where σ_k is the kth singular value, with singular values in the decreasing order, and $[u_{1k}, u_{2k}, \cdots, u_{Ik}]^T$ and $[v_{1k}, v_{2k}, \cdots, v_{Jk}]^T$ are the kth left and right singular vectors, respectively. When $\boldsymbol{X}$ has full rank, the maximum dimensionality is $\min(I-1, J-1)$, where the "-1" is due to the subtraction of elements $p_{i+}p_{+j}$, that leads to a centering of the elements of $\boldsymbol{X}$ as $\sum_i (p_{ij} - p_{i+}p_{+j}) = 0 = \sum_j (p_{ij} - p_{i+}p_{+j})$. Multiplying the singular vectors consisting of elements u_{ik} and v_{jk} by $p_{i+}^{-\frac{1}{2}}$ and $p_{+j}^{-\frac{1}{2}}$, respectively, leads to

$$\frac{p_{ij}}{p_{i+}p_{+j}} - 1 = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k \phi_{ik} \gamma_{jk}, \tag{3}$$

where $\phi_{ik} = p_{i+}^{-\frac{1}{2}} u_{ik}$ and $\gamma_{jk} = p_{+j}^{-\frac{1}{2}} v_{jk}$. Scores $\phi_{ik}, k = 1, 2, \cdots, K$ and $\gamma_{jk}, k = 1, 2, \cdots, K$ provide the standard coordinates of row point i and column point j in K-dimensional space, respectively, because of $\sum_i p_{i+}\phi_{ik} = \sum_j p_{+j}\gamma_{jk} = 0$ and $\sum_i p_{i+}\phi_{ik}^2 = \sum_j p_{+j}\gamma_{jk}^2 = 1$. Scores $\phi_{ik}\sigma_k, k = 1, 2, \cdots, K$ and $\gamma_{jk}\sigma_k, k = 1, 2, \cdots, K$ provide the principle coordinates of row point i and column point j in K-dimensional space, respectively. When $K < \min(I-1, J-1)$, the Euclidean distances between these row (column) points approximate so-called χ^2-distances between rows (columns) of $\boldsymbol{X}$. The squared χ^2-distance between rows i and i' of $\boldsymbol{X}$ is

$$\delta_{ii'}^2 = \sum_j \frac{\left(\frac{p_{ij}}{p_{i+}} - \frac{p_{i'j}}{p_{i'+}}\right)^2}{p_{+j}}, \tag{4}$$

and similarly for the chi-squared distance between columns j and j'. Equation (4) shows that the χ^2-distance $\delta_{ii'}$ measures the difference between the ith vector of conditional proportions p_{ij}/p_{i+} and the i'th vector of conditional proportions $p_{i'j}/p_{i'+}$, where more weight is given to the differences in elements j if p_{+j} is relatively smaller compared to other columns.

Although the use of Euclidean distance is standard in CA, Qi, Hessen, and Van der Heijden (2023) show that for information retrieval cosine similarity leads to the best performance among Euclidean distance, dot similarity, and cosine similarity. The superiority of cosine similarity also holds in the context of word embedding studies (Bullinaria & Levy, 2007). Therefore, in this paper we use cosine similarity to calculate the similarity of row points and of column points. It is worth noting that $p_{i+}^{-\frac{1}{2}}$ in $\phi_{ik} = p_{i+}^{-\frac{1}{2}} u_{ik}$ and $p_{+j}^{-\frac{1}{2}}$ in $\gamma_{jk} = p_{+j}^{-\frac{1}{2}} v_{jk}$ have no effects on the cosine similarity. Details are in Supplementary materials A. We coin scores $u_{ik}\sigma_k, k = 1, 2, \cdots, K$ and $v_{jk}\sigma_k, k = 1, 2, \cdots, K$ an alternative coordinates system for CA directly suited for cosine similarity.

The so-called total inertia is

$$\sum_i \sum_j \frac{(p_{ij} - p_{i+}p_{+j})^2}{p_{i+}p_{+j}} = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k^2. \tag{5}$$

This illustrates that CA decomposes the total inertia over $\min(I-1, J-1)$ dimensions. The total inertia equals the well-known Pearson χ^2 statistic divided by x_{++}, so that the total inertia does not depend on the sample size x_{++}. The relative contribution of cell (i, j) to the total inertia is calculated as $\frac{(p_{ij}-p_{i+}p_{+j})^2}{p_{i+}p_{+j}} / \sum_i \sum_j \frac{(p_{ij}-p_{i+}p_{+j})^2}{p_{i+}p_{+j}}$. The relative contribution of the ith row (jth column) to the kth dimension is calculated as u_{ik}^2 (v_{jk}^2).

3.2 The objective function of CA

To simplify the later comparison of CA with the other models, we present the objective function that is minimized in CA. The objective function is (Greenacre, 1984, pp. 345-349):

$$\sum_{i,j} p_{i+}p_{+j} \left(\frac{p_{ij}}{p_{i+}p_{+j}} - 1 - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2, \tag{6}$$

where $\boldsymbol{e}_i$ and $\boldsymbol{o}_j$ are parameter vectors for target word i and context word j, with respect to which the objective function is minimized. The vectors have length $K \leq \min(I-1, J-1)$. We call the part of the formula to be approximated, i.e. $(p_{ij}/p_{i+}p_{+j} - 1)$, the fitting function and the weighting part $p_{i+}p_{+j}$ the weighting function. Thus, according to (6), CA can be viewed as a weighted matrix factorization of $(p_{ij}/p_{i+}p_{+j} - 1)$ with weighting function

$p_{i+}p_{+j}$.

The solution is found using the SVD as in Equation (2). The K-dimensional approximation of $(p_{ij}/p_{i+}p_{+j} - 1)$ is

$$\frac{p_{ij}}{p_{i+}p_{+j}} - 1 \approx \sum_{k=1}^{K} \sigma_k \phi_{ik} \gamma_{jk} = \boldsymbol{e}_i^T \boldsymbol{o}_j. \tag{7}$$

The matrix $[\boldsymbol{e}_i^T \boldsymbol{o}_j]$ minimizes (6) amongst all matrices of rank K in a weighted least-squares sense (Greenacre, 1984). The parameter vectors $\boldsymbol{e}_i$ and $\boldsymbol{o}_j$ can be represented, for example, as

$$\boldsymbol{e}_i = [\phi_{i1}\sigma_1, \phi_{i2}\sigma_2, \cdots, \phi_{iK}\sigma_K]^T \tag{8}$$

and

$$\boldsymbol{o}_j = [\gamma_{j1}, \gamma_{j2}, \cdots, \gamma_{jK}]^T \tag{9}$$

As described above, this representation $\boldsymbol{e}_i$ of target word i has the advantage that the χ^2-distance between target words i and i' in the original matrix is approximated by the Euclidean distance between $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i'}$.

The parameters $\boldsymbol{e}_i$ can be adjusted by a singular value weighting exponent p, i.e., $\boldsymbol{e}_i = [\phi_{i1}\sigma_1^p, \phi_{i2}\sigma_2^p, \cdots, \phi_{iK}\sigma_K^p]^T$. Correspondingly, the alternative coordinate for the adjusted row i by a singular value weighting exponent is $[u_{i1}\sigma_1^p, u_{i2}\sigma_2^p, \cdots, u_{iK}\sigma_K^p]^T$.

3.3 Three variants of CA for word embeddings

We present three variants of CA. According to Stratos et al. (2015), word counts can be naturally modeled as Poisson variables. The square-root transformation of a Poisson variable leads to stabilization of the variance (Bartlett, 1936; Stratos et al., 2015). Stratos et al. (2015) proposed to combine CCA with the square-root transformation of the word-context matrix. Even though CA of a contingency table is equivalent to CCA of the data in the form of an indicator matrix, we call the proposal by Stratos et al. (2015) ROOT-CCA, to distinguish it from the alternative ROOT-CA, discussed later.

ROOT-CCA In ROOT-CCA, an SVD is performed on the matrix whose typical element is the square root of $x_{ij}/\sqrt{x_{i+}x_{+j}} = p_{ij}/\sqrt{p_{i+}p_{+j}}$, that is

$$\sqrt{\frac{p_{ij}}{\sqrt{p_{i+}p_{+j}}}} = \sum_{k=1}^{\min(I,J)} \sigma_k u_{ik} v_{jk}. \tag{10}$$

The reason that Stratos et al. (2015) ignore $p_{i+}p_{+j}$ in $p_{ij} - p_{i+}p_{+j}$ (compare Equation (2)) is that they believe that, when the sample size x_{++} is large, the first part $p_{ij}/\sqrt{(p_{i+}p_{+j})}$ in $(p_{ij} - p_{i+}p_{+j})/\sqrt{(p_{i+}p_{+j})}$ dominates the expression.

ROOT-CA Inspired by Stratos et al. (2015), we present CA of the square-root transformation of the word-context matrix (ROOT-CA) (Bartlett, 1936; Hsu & Culhane, 2023). ROOT-

CA differs from ROOT-CCA in the following way. In the ROOT-CA, first we create a square-root transformation of the word-context matrix with elements $\sqrt{x_{ij}}$, and then we perform CA on this matrix. Let $p^*_{ij} = \frac{\sqrt{x_{ij}}}{\sum_{ij}\sqrt{x_{ij}}} = \frac{\sqrt{p_{ij}}}{\sum_{ij}\sqrt{p_{ij}}}$. Then ROOT-CA provides the decomposition

$$\frac{p^*_{ij} - p^*_{i+}p^*_{+j}}{\sqrt{p^*_{i+}p^*_{+j}}} = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k u_{ik} v_{jk}. \tag{11}$$

ROOTROOT-CA According to Stratos et al. (2015), word counts can be naturally modeled as Poisson variables. In the Poisson distribution the mean and variance are identical. The phenomenon of the data having greater variability than expected based on a statistical model is called overdispersion (Agresti, 2007). In the context of CA, Nishisato et al. (2021) point out, generally speaking, a two-way contingency table is prone to overdispersion. Overdispersion may negatively affect the performance of CA (Beh et al., 2018; Nishisato et al., 2021).

Greenacre (2009, 2010), referring to Field et al. (1982), points out that in ecology abundance data is almost always highly over-dispersed and a particular school of ecologists routinely applies a fourth-root transformation before proceeding with the statistical analysis. Therefore we also study the effect of a root-root transformation before performing CA. We call it ROOTROOT-CA. That is, ROOTROOT-CA is a CA on the matrix with typical element $\sqrt{\sqrt{x_{ij}}}$ (Field et al., 1982). Suppose $p^{**}_{ij} = \frac{\sqrt{\sqrt{x_{ij}}}}{\sum_{ij}\sqrt{\sqrt{x_{ij}}}} = \frac{\sqrt{\sqrt{p_{ij}}}}{\sum_{ij}\sqrt{\sqrt{p_{ij}}}}$. Then, we have

$$\frac{p^{**}_{ij} - p^{**}_{i+}p^{**}_{+j}}{\sqrt{p^{**}_{i+}p^{**}_{+j}}} = \sum_{k=1}^{\min(I-1,J-1)} \sigma_k u_{ik} v_{jk}, \tag{12}$$

Thus ROOT-CA and ROOTROOT-CA are pre-transformations of the elements x_{ij} of the original matrix by $\sqrt{x_{ij}}$ and $\sqrt{\sqrt{x_{ij}}}$, respectively. CA is performed on the transformed matrix.

3.4 T-Test

The T-Test (TTEST) weighting scheme, described by Curran and Moens (2002) and Curran (2004), focuses on the matrix of standardized residuals, see Equation (1). Thus it is remarkably similar to CA, where the matrix of standardized residuals is decomposed. For a comparison between CA and TTEST weighting in word similarity tasks, as we will carry out below, the question is whether the performance is better on the matrix of standardized residuals, or on a low dimensional representation of this matrix provided by CA.

Inspired by Section 3.3, we also explore the performances of the matrix STRATOS-TTEST with typical element $\sqrt{p_{ij}/\sqrt{p_{i+}p_{+j}}}$ (compare Equation (10)), the matrix ROOT-TTEST with typical element $\left(p^*_{ij} - p^*_{i+}p^*_{+j}\right)/\sqrt{p^*_{i+}p^*_{+j}}$ (compare Equation (11)), and the matrix ROOTROOT-TTEST with typical element $\left(p^{**}_{ij} - p^{**}_{i+}p^{**}_{+j}\right)/\sqrt{p^{**}_{i+}p^{**}_{+j}}$ (compare Equation (12)).

4 PMI-based word embedding methods

4.1 PMI-SVD and PPMI-SVD

Pointwise mutual information (PMI) is an important concept in NLP. The PMI between a target word i and a context word j is defined as (Bullinaria & Levy, 2007; Jurafsky & Martin, 2023; Levy & Goldberg, 2014; Levy et al., 2015):

$$\text{PMI}(i, j) = \log \frac{p_{ij}}{p_{i+}p_{+j}} \tag{13}$$

i.e. the log of the contingency ratios (Greenacre, 2009, 2017), also known as Pearson ratios (Beh & Lombardo, 2021; Goodman, 1996), $p_{ij}/\left(p_{i+}p_{+j}\right)$. If $p_{ij} = 0$, then $\text{PMI}(i, j) = \log 0 = -\infty$, and it is usual to set $\text{PMI}(i, j) = 0$ in this situation.

A common approach is to factorize the PMI matrix using SVD, which we call PMI-SVD. Thus the objective function is

$$\sum_{i,j} \left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \boldsymbol{e}_i^T \boldsymbol{o}_j \right)^2 . \tag{14}$$

In terms of a weighted matrix factorization, PMI-SVD is the matrix factorization of the PMI matrix with the weighting function 1. The solution is provided directly via SVD. An SVD applied to the PMI matrix with elements $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ yields

$$\log \frac{p_{ij}}{p_{i+}p_{+j}} = \sum_{k=1}^{\min(I,J)} \sigma_k u_{ik} v_{jk}, \tag{15}$$

where $\min(I, J)$ is the rank of the PMI matrix. The K-dimensional approximation of $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ is

$$\log \frac{p_{ij}}{p_{i+}p_{+j}} \approx \sum_{k=1}^{K} \sigma_k u_{ik} v_{jk} = \boldsymbol{e}_i^T \boldsymbol{o}_j \tag{16}$$

where the matrix with elements $\boldsymbol{e}_i^T \boldsymbol{o}_j$ minimizes (14) amongst all matrices of rank K in a least squares sense, where $K \leq \min(I, J)$. Both CA and PMI-SVD are dimensionality reduction techniques making use of SVD.

The parameters $\boldsymbol{e}_i$ and $\boldsymbol{o}_j$ can be represented as

$$\boldsymbol{e}_i = \left[u_{i1}\sigma_1, u_{i2}\sigma_2, \cdots, u_{iK}\sigma_K\right]^T \tag{17}$$

and

$$\boldsymbol{o}_j = \left[v_{j1}, v_{j2}, \cdots, v_{jK}\right]^T . \tag{18}$$

Thus the Euclidean distance between target words i and i' in the original matrix is approximated by the Euclidean distance between $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i'}$. In practice, one regularly sees that the parameters $\boldsymbol{e}_i$ are adjusted by an exponent p used for weighting the singular values, i.e., $\boldsymbol{e}_i = \left[u_{i1}\sigma_1^p, u_{i2}\sigma_2^p, \cdots, u_{iK}\sigma_K^p\right]^T$, where p is usually set to 0 or 0.5 (Levy & Goldberg, 2014; Levy et al., 2015; Stratos et al., 2015).

It is worth noting that the elements in the PMI matrix, where word-context pairs that co-occur rarely are negative, but word-context pairs that never co-occur are set to 0 (Levy & Goldberg, 2014), are not monotonic transformations of observed counts divided by counts under independence. For this reason an alternative is proposed, namely the positive PMI matrix, abbreviated as PPMI matrix. In the PPMI matrix all negative values are set to 0:

$$\text{PPMI}(i, j) = \max\left(\text{PMI}(i, j), 0\right) \tag{19}$$

In most applications, one makes use of the PPMI matrix instead of the PMI matrix (Salle et al., 2016). We call the factorization of the PPMI matrix using SVD PPMI-SVD (Zhang et al., 2022).

4.2 GloVe

The GloVe objective function to be minimized is (Pennington et al., 2014):

$$\sum_{i,j} f(x_{ij}) \left(\log x_{ij} - b_i - s_j - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2 \tag{20}$$

where

$$f(x_{ij}) = \begin{cases} (x_{ij}/x_{\max})^{\alpha} & \text{if } x_{ij} < x_{\max} \\ 1 & \text{otherwise} \end{cases}$$

In addition to parameter vectors $\boldsymbol{e}_i$ and $\boldsymbol{o}_j$, the scalar parameter terms b_i and s_j are referred to as $bias$ of target word i and context word j, respectively. Pennington et al. (2014) train the GloVe model using an adaptive gradient algorithm (AdaGrad) (Duchi et al., 2011). This algorithm trains only on the non-zero elements of a word-context matrix, as $f(0) = 0$, which avoids the appearance of the undefined $\log 0$ in Equation (20).

In the original proposal of GloVe (Pennington et al., 2014), $b_i = \log x_{i+}$ and then, due to the symmetric role of target word and context word, $s_j = \log x_{+j}$. Shi and Liu (2014) and Shazeer et al. (2016) show that the bias terms b_i and s_j are highly correlated with $\log x_{i+}$ and $\log x_{+j}$, respectively, in GloVe model training. This means that the GloVe model minimizes a weighted least squares loss function with the weighting function $f(x_{ij})$ and approximate fitting function $\log x_{ij} - \log x_{i+} - \log x_{+j} = \log\left(x_{ij}x_{++}/\left(x_{i+}x_{+j}\right)\right) - \log x_{++} = \log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right) - \log x_{++}$:

$$\sum_{i,j} f(x_{ij}) \left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \log x_{++} - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2 \tag{21}$$

4.3 Skip-gram with negative sampling

SGNS stands for skip-gram with negative sampling of word2vec embeddings (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013). The algorithms used in SGNS are stochastic gradient descent and backpropagation (Rong, 2014; Rumelhart, Hinton, & Williams, 1986). SGNS trains word embeddings on every word of the corpus one by one.

Levy and Goldberg (2014) showed that SGNS implicitly factorizes a PMI matrix shifted by log n:

$$\log \frac{p_{ij}}{p_{i+}p_{+j}} - \log n \approx \boldsymbol{e}_i^T \boldsymbol{o}_j \tag{22}$$

where n is the number of negative samples. According to Levy and Goldberg (2014) and Shazeer et al. (2016), the objective function of SGNS is approximately a minimization of the difference between $\boldsymbol{e}_i^T \boldsymbol{o}_j$ and $\log\ (p_{ij}/\ (p_{i+}p_{+j}) - \log n)$, tempered by a monotonically increasing weighting function of the observed co-occurrence count x_{ij}, that we denote by $g(x_{ij})$:

$$\sum_{i,j} g(x_{ij}) \left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \log n - \boldsymbol{e}_i^T \boldsymbol{o}_j \right)^2 \tag{23}$$

This shows that SGNS differs from GloVe in the use of n instead of x_{++}, and $g(x_{ij})$ instead of $f(x_{ij})$.

5 Relationships of CA to PMI-based models

5.1 CA and PMI-SVD / PPMI-SVD

In this section, we discuss PMI-SVD and PPMI-SVD together, as PMI and PPMI are the same except that in PPMI all negative values of PMI are set to 0.

CA is closely related to PMI-SVD. This becomes clear by comparing $(p_{ij}/\ (p_{i+}p_{+j}) - 1)$ in (6) with PMI$(i,j) = \log (p_{ij}/\ (p_{i+}p_{+j}))$ in (14). The relation lies in a Taylor expansion of $\log (p_{ij}/\ (p_{i+}p_{+j}))$, namely that, if x is small, $\log(1+x) \approx x$ (Van der Heijden, De Falguerolles, & De Leeuw, 1989). Substituting x with $p_{ij}/(p_{i+}p_{+j}) - 1$ leads to:

$$\log \frac{p_{ij}}{p_{i+}p_{+j}} \approx \frac{p_{ij}}{p_{i+}p_{+j}} - 1 \tag{24}$$

This illustrates that if $(p_{ij}/p_{i+}p_{+j} - 1)$ is small, the objective function of CA approximates

$$\sum_{i,j} p_{i+}p_{+j} \left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \boldsymbol{e}_i^T \boldsymbol{o}_j \right)^2 . \tag{25}$$

From Equation (25) it follows that CA is approximately a weighted matrix factorization of $\log\ (p_{ij}/\ (p_{i+}p_{+j}))$ with weighting function $p_{i+}p_{+j}$. The Equation (24) can also be obtained by the Box-Cox transformation of the contingency ratios, for example, Greenacre (2009) and Beh and Lombardo (2024), and we refer to their work for more details.

Comparing Equation (25) with Equation (14), both CA and PMI-SVD can be taken as weighted least squares methods having approximately the same fitting functions, namely $(p_{ij}/p_{i+}p_{+j} - 1)$ for CA and $\log (p_{ij}/\ (p_{i+}p_{+j}))$ for PMI-SVD. Both make use of an SVD.

However, they use different weighting functions, namely $p_{i+}p_{+j}$ in CA and 1 in PMI-SVD. It has been argued that equally weighting errors in the objective function, as is the case in PMI-SVD, is not a good approach (Salle & Villavicencio, 2023; Salle et al., 2016). For example, Salle and Villavicencio (2023) presented the reliability principle, that the objective function should have a weight on the reconstruction error that is a monotonically increasing function of the marginal frequencies of word and of context. On the other hand, CA, unlike PMI-SVD, weights errors in the objective function with a weighting function equal to the product of the marginal proportions of word and context (Beh & Lombardo, 2021; Greenacre, 1984, 2017).

5.1.1 PMI-GSVD

The weighting function of PMI-SVD is 1 while in the approximate version of CA it is $p_{i+}p_{+j}$. Therefore, we also investigate the performance of a weighted factorization of the PMI matrix, where $p_{i+}p_{+j}$ is the weighting function:

$$\sum_{i,j} p_{i+}p_{+j}\left(\log \frac{p_{ij}}{p_{i+}p_{+j}} - \boldsymbol{e}_i^T \boldsymbol{o}_j\right)^2. \tag{26}$$

Similar with CA, we use generalized SVD (GSVD) to find the optimum of the objective function (PMI-GSVD). That is, an SVD is applied as follows:

$$\sqrt{p_{i+}p_{+j}}\log \frac{p_{ij}}{p_{i+}p_{+j}} = \sum_{k=1}^{\min(I,J)} \sigma_k u_{ik} v_{jk}, \tag{27}$$

We call the matrix with typical element $\sqrt{p_{i+}p_{+j}}\log \frac{p_{ij}}{p_{i+}p_{+j}}$ the WPMI matrix, also known as the modified log-likelihood ratio residual (Beh & Lombardo, 2024).

5.2 CA and GloVe

Both CA and GloVe are weighted least squares methods. The weighting function in GloVe is $f(x_{ij})$, which is defined uniquely for each element of the word-context matrix, while the weighting function $p_{i+}p_{+j}$ in CA is defined by the row and column margins.

In the approximate fitting function of GloVe, $\log(p_{ij}/(p_{i+}p_{+j})) - \log x_{++}$, the term $\log x_{++}$ can be considered as a shift of $\log(p_{ij}/(p_{i+}p_{+j}))$. And as we showed in Section 5.1, the fitting function of CA is approximately $\log(p_{ij}/(p_{i+}p_{+j}))$ when p_{ij} is close to $p_{i+}p_{+j}$. Thus, from a comparison of the objective functions of CA and GloVe, it is natural to expect that these two methods will yield similar results if $(p_{ij}/p_{i+}p_{+j} - 1)$ is small.

In comparing the algorithms of these two methods, we find that CA uses SVD while GloVe uses AdaGrad. These two algorithms have their own advantages and disadvantages. On the one hand, the AdaGrad algorithm trains GloVe only on the nonzero elements of word-context matrix, one by one, while in CA the SVD decomposes the entire word-context matrix in full in one step. On the other hand, the SVD always finds the global minimum while the AdaGrad algorithm cannot guarantee the global minimum.

5.3 CA and SGNS

By comparing Equations (23) and (25), both the approximation of CA and of SGNS are found by weighted least squares methods. The weighting function in SGNS is $g(x_{ij})$, which is defined for each element of word-context matrix where frequent word-context pairs pay more for deviations than infrequent ones (Levy & Goldberg, 2014), while the weighting function in CA is defined by the row and column margins, i.e. $p_{i+}p_{+j}$.

In the fitting function of the approximation of SGNS, $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right) - \log n$, the term $\log n$ can be considered as a shift of $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$. As shown in Section 5.1, the approximate fitting function in CA is $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$. Thus, considering the objective function view, both the approximation of CA and of SGNS make use of the PMI matrix.

Although the approximate objective function of SGNS is similar to that of CA, the training processing for SGNS is different from that of CA. SGNS trains word embeddings on the words of a corpus, one by one, to maximize the probabilities of target words and context words co-occurrence, and to minimize the probabilities between target words and randomly sampled words, by updating the vectors of target words and context words. In contrast, CA first counts all co-occurrences in the corpus and then performs SVD on the matrix of standardized residuals to obtain the vectors of target words and context words at once.

6 Two corpora and five word similarity datasets

All methods are trained on two corpora: Text8 (*Text8 dataset*, 2006) and British National Corpus (BNC) (BNC Consortium, 2007), respectively. Text8 is a widely used corpus in NLP (Guo & Yao, 2021; Podkorytov, Biś, Cai, Amirizirtol, & Liu, 2020; Roesler, Aly, Taniguchi, & Hayashi, 2019; Xin, Yuan, He, & Jose, 2018). It includes more than 17 million words from Wikipedia (Peng & Feldman, 2017) and only consists of lowercase English characters and spaces. Words that appeared less than 100 times in the corpus are ignored, resulting in a vocabulary of 11,815 terms.

BNC is from a representative variety of sources and is widely used (Raphael, 2023; Samuel, Kutuzov, Øvrelid, & Velldal, 2023). Data cited herein have been extracted from the British National Corpus, distributed by the University of Oxford on behalf of the BNC Consortium. We remove English punctuation and numbers and set words in lowercase form. Words that appeared less than 500 times in the corpus are ignored, resulting in a vocabulary of 11,332 terms.

Following previous studies (Levy et al., 2015; Pakzad & Analoui, 2021), we evaluate each word embeddings method on word similarity tasks using the Spearman's correlation coefficient ρ. We use five popular word similarity datasets: WordSim353 (Finkelstein et al., 2002), MEN (Bruni, Boleda, Baroni, & Tran, 2012), Mechanical Turk (Radinsky, Agichtein, Gabrilovich, & Markovitch, 2011), Rare (Luong, Socher, & Manning, 2013), and SimLex-

999 (Hill, Reichart, & Korhonen, 2015). All these datasets consist of word pairs together with human-assigned similarity scores. For example, in WordSim353, where scores range from 0 (least similar) to 10 (most similar), one word pair is (tiger, cat) with human assigned similarity score 7.35. Out-of-vocabulary words are removed from all test sets. I.e., if either tiger or cat doesn't occur in the vocabularies of the 11,815 terms created by Text8 corpus, we delete (tiger, cat). Thus for evaluating the different word embedding methods in Text8 277 word pairs with scores are kept in WordSim353 instead of the original 353 word pairs. Table 1 provides the number of word pairs used by the datasets in Text8 and BNC.

Table 1: Datasets for word similarity evaluation.

Dataset	Word pairs	Word pairs in Text8	Word pairs in BNC
WordSim353	353	277	276
MEN	3000	1544	1925
Turk	287	221	197
Rare	2034	205	204
SimLex-999	999	726	847

After calculating the solutions for CA, PMI-SVD, PPMI-SVD, PMI-GSVD, ROOT-CA, ROOTROOT-CA, ROOT-CCA, GloVe, and SGNS, we obtain the word embeddings. We calculate the cosine similarity for each word pair in each word similarity dataset. For example, for WordSim353 using Text8, we obtain 277 cosine similarities. The Spearman's correlation coefficient ρ (Hollander, Wolfe, & Chicken, 2013) between these similarities and the human similarity scores is calculated to evaluate these word embedding methods. Larger values are better.

7 Study setup

7.1 SVD-based methods

CA, PMI-SVD, PPMI-SVD, PMI-GSVD, ROOT-CA, ROOTROOT-CA, and ROOT-CCA are SVD-based dimensionality reduction methods. First, we create a word-context matrix of size 11,815×11,815 and 11,332×11,332 based on Text8 and BNC, respectively. We use a window of size 2, i.e., two words to each side of the target word. A context word one token and two tokens away will be counted as $1/1$ and $1/2$ of an occurrence, respectively. Then we perform SVD on the related matrices. We use the svd function from scipy.linalg in Python to calculate the SVD of a matrix, and obtain singular values σ_k, left singular vectors u_{ik}, and right singular vectors v_{jk}. We obtain the word embeddings as $\boldsymbol{e}_i = \left[u_{i1}\sigma_1^p, u_{i2}\sigma_2^p, \cdots, u_{ik}\sigma_k^p\right]^T$.

The choices of the exponent weighting p and number of dimensions k are important for SVD-based methods. In the context of PPMI-SVD and ROOT-CCA p is regularly set to $p = 0$ or $p = 0.5$ (Levy & Goldberg, 2014; Levy et al., 2015; Stratos et al., 2015). For $p = 0$, we have the standard coordinates with $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$. For $p = 0.5$, we have $\boldsymbol{A}_k = \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^T = (\boldsymbol{U}_k\boldsymbol{\Sigma}_k^{1/2})(\boldsymbol{V}_k\boldsymbol{\Sigma}_k^{1/2})^T$. That is, the target words $\boldsymbol{U}_k\boldsymbol{\Sigma}_k^{1/2}$ and context words $\boldsymbol{V}_k \boldsymbol{\Sigma}_k^{1/2}$ reconstruct the decomposed matrix $\boldsymbol{A}_k$. The two created word-context matrices based on Text8 and BNC are symmetric, so the matrices to be decomposed are also symmetric. For the SVD of a symmetric matrix, using the target words $\boldsymbol{U}_k \boldsymbol{\Sigma}_k^{1/2}$ for word embeddings is equivalent to using the context words $\boldsymbol{V}_k \boldsymbol{\Sigma}_k^{1/2}$ for word embeddings. We vary the number of dimensions k from 2, 50, 100, 200, $\cdots$, 1,000, 2,000, $\cdots$, 10,000.

7.2 GloVe and SGNS

We use the public implementation by Pennington et al. (2014) to perform GloVe and choose the default hyperparameters. Pennington et al. (2014) proposed to use the context vectors $\boldsymbol{o}_j$ in addition to target word vectors $\boldsymbol{e}_i$. Here, we only use target word vectors $\boldsymbol{e}_i$, set window size to 2 and set vocab minimum count to 100 for Text8 and 500 for BNC, in the same way as for the SVD-based methods to keep the settings consistent. We vary the dimension k of word embeddings from 200 to 600 with intervals of 100.

We use the public implementation by Mikolov, Sutskever, et al. (2013) to perform SGNS, and use the vocabulary created by GloVe as the input of SGNS. We choose the default values except for the dimensions k of word embeddings and window size, which are chosen in the same way as in GloVe, to keep the settings consistent.

8 Results

We make a distinction between conditions where no dimensionality reduction takes place, and conditions where dimensionality reduction is used. For no dimensionality reduction we compare TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, STRATOS-TTEST. For dimensionality reduction we first compare CA with the more standard methods PMI-SVD, PPMI-SVD, PMI-GSVD, GloVe, SGNS, and then compare variants of CA.

8.1 TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST

First, we compare methods where no dimensionality reduction takes place. We show the Spearman's correlation coefficient ρ for the TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST matrices in Table 2. The results for the five word similarity datasets and the two corpora show that (1) either ROOT-TTEST or ROOTROOT-TTEST is best, and (2) ROOT-TTEST is consistently better than PPMI, PMI, STRATOS-TTEST, and WPMI. In the Total column of the block at the bottom of the table we provide the sum of ρ-values for all five datasets and two corpora. Overall, ROOT-TTEST and ROOTROOT-TTEST perform best, closely followed by PPMI and TTEST. PMI follows at some distance, and last, we find STRATOS-TTEST and WPMI.

Table 2: Correlation coefficient ρ for fitting matrix.

		Text8	BNC	Total
WordSim353	TTEST	0.588	0.427	1.015
	PMI	0.587	0.292	0.879
	PPMI	0.609	0.505	1.115
	WPMI	0.233	0.221	0.454
	ROOT-TTEST	**0.658**	**0.539**	**1.197**
	ROOTROOT-TTEST	0.646	0.495	1.141
	STRATOS-TTEST	0.438	0.314	0.752
MEN	TTEST	0.248	0.260	0.509
	PMI	0.269	0.224	0.494
	PPMI	0.253	0.284	0.537
	WPMI	0.132	0.171	0.303
	ROOT-TTEST	0.305	**0.293**	**0.598**
	ROOTROOT-TTEST	**0.317**	0.263	0.580
	STRATOS-TTEST	0.156	0.130	0.286
Turk	TTEST	0.619	0.649	1.268
	PMI	0.629	0.514	1.143
	PPMI	0.651	0.625	1.276
	WPMI	0.343	0.417	0.760
	ROOT-TTEST	0.666	**0.659**	**1.325**
	ROOTROOT-TTEST	**0.667**	0.616	1.283
	STRATOS-TTEST	0.561	0.525	1.086
Rare	TTEST	0.392	0.428	0.820
	PMI	0.335	0.289	0.624
	PPMI	0.328	0.363	0.691
	WPMI	0.252	0.255	0.506
	ROOT-TTEST	0.389	**0.477**	0.866
	ROOTROOT-TTEST	**0.418**	0.454	**0.872**
	STRATOS-TTEST	0.243	0.196	0.439
SimLex-999	TTEST	0.220	0.230	0.450
	PMI	0.257	0.168	0.425
	PPMI	0.251	0.277	0.528
	WPMI	0.139	0.118	0.257
	ROOT-TTEST	**0.276**	**0.280**	**0.556**
	ROOTROOT-TTEST	0.271	0.239	0.509
	STRATOS-TTEST	0.181	0.125	0.306
Total	TTEST	2.067	1.994	4.061
	PMI	2.078	1.487	3.565
	PPMI	2.092	2.054	4.146
	WPMI	1.098	1.182	2.280
	ROOT-TTEST	2.293	**2.249**	**4.542**
	ROOTROOT-TTEST	**2.319**	2.067	4.386
	STRATOS-TTEST	1.579	1.289	2.869

8.2 CA, PMI-SVD, PPMI-SVD, PMI-GSVD, GloVe, and SGNS

Next, we compare CA (RAW-CA in Table 3) with the PMI-based methods PMI-SVD, PPMI-SVD, PMI-GSVD, GloVe, and SGNS. Table 3 has a left part, where $p = 0$, and a right part, where $p = 0.5$. As p does not exist in GloVe and SGNS, these methods have identical values for $p = 0$ and $p = 0.5$. Plots for ρ as a function of k for SVD-based methods are in Supplementary materials B.

Comparing the last block of Table 3 with the last block of Table 2 reveals that, overall, dimensionality reduction is beneficial for the size of ρ, as CA, PMI-SVD, PPMI-SVD, PMI-GSVD, ROOT-CA, ROOTROOT-CA, and ROOT-CCA do better than their respective counterparts TTEST, PMI, PPMI, WPMI, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST. For TTEST the improvement due to using SVD is less than for PMI, PPMI, WPMI, ROOT-TTEST, STRATOS-TTEST; for WPMI and STRATOS-TTEST the improvement due to using SVD is more than for TTEST, PPMI, ROOT-TTEST, and ROOTROOT-TTEST, which is a result consistent for each corpus and each word similarity dataset.

For an overall comparison of the dimensionality reduction methods, we study the block at the bottom of Table 3, which provides the sum of the ρ-values over the five word similarity datasets. For both $p = 0$ and $p = 0.5$, among RAW-CA, PMI-SVD, PPMI-SVD, PMI-GSVD, GloVe, and SGNS, overall PMI-SVD and PPMI-SVD perform best, closely followed by SGNS. RAW-CA and PMI-GSVD follow at some distance, and last, we find GloVe. The popular method GloVe does not perform well. Possibly the conditions of the study are not optimal for GloVe, as the Text8 and BNC corpora are, with 11,815 and 11,332 terms respectively, possibly too small to obtain reliable results (Jiang, Yu, Hsieh, & Chang, 2018).

As the focus in this paper is on the performance of CA, we give some extra attention to RAW-CA and the similar PMI-GSVD. Even though CA and PMI-GSVD have the same weighting function $p_{i+}p_{+j}$, and should be close when $p_{ij}/\left(p_{i+}p_{+j}\right) - 1$ is small (compare the discussion around Equations (24, 25)) their performances are rather different. This may be because there are extremely large values (larger than 35,000) in the fitting function $\left(p_{ij}/\left(p_{i+}p_{+j}\right) - 1\right)$ of CA, which makes the fitting function of CA not close to the fitting function $\log\left(p_{ij}/\left(p_{i+}p_{+j}\right)\right)$ of PMI-GSVD.

When we compare PMI-GSVD with PMI-SVD, we are surprised to find that weighting rows and columns appears to decrease the values of ρ. This is in contrast with the reliability principle of Salle and Villavicencio (2023) discussed above.

We now discuss why PMI-SVD and PPMI-SVD do better than PMI-GSVD. It turns out that the number and sizes of extreme values in the matrix WPMI decomposed by PMI-GSVD are much larger than in PMI and PPMI, and this results in PMI-GSVD dimensions being dominated by single words. We only include non-zero elements in the PMI matrix as the PMI matrix is sparse: 94.2% of the entries are zero for Text8; for a fair comparison, the corresponding 94.2% of entries in the PPMI and WPMI matrices are also ignored. Following box plot methodology (Dodge, 2008; Schwertman, Owens, & Adnan, 2004; Tukey, 1977), extreme values are determined as follows: let q_1 and q_3 be the first and third sample quartiles, and let $f_1 = q_1 - 1.5(q_3 - q_1)$, $f_3 = q_3 + 1.5(q_3 - q_1)$. Then extreme values are de-

Table 3: SVD: correlation coefficient ρ with $p = 0, 0.5$.

		$p = 0$					$p = 0.5$				
		Text8		BNC			Text8		BNC		
		k	ρ	k	ρ	total	k	ρ	k	ρ	total
WordSim353	RAW-CA	600	0.578	400	0.465	1.043	9000	0.609	10000	0.498	1.107
	PMI-SVD	400	0.675	600	0.628	1.303	400	0.683	500	0.579	1.262
	PPMI-SVD	400	0.681	700	0.628	1.309	200	0.694	2000	0.623	1.317
	GloVe	200	0.422	600	0.522	0.943	200	0.422	600	0.522	0.943
	SGNS	300	0.668	600	0.551	1.219	300	0.668	600	0.551	1.219
	PMI-GSVD	700	0.512	600	0.468	0.980	6000	0.548	3000	0.449	0.997
	ROOT-CA	300	0.668	400	0.623	1.291	500	0.688	900	**0.657**	**1.345**
	ROOTROOT-CA	200	**0.692**	200	**0.635**	**1.327**	300	**0.697**	400	0.630	1.327
	ROOT-CCA	100	0.682	700	0.627	1.310	300	0.684	600	0.620	1.304
MEN	RAW-CA	300	0.223	600	0.293	0.516	7000	0.256	9000	0.299	0.556
	PMI-SVD	800	0.328	700	0.393	0.721	600	0.317	2000	0.357	0.674
	PPMI-SVD	800	0.336	500	0.394	0.730	800	0.324	1000	0.358	0.681
	GloVe	300	0.175	600	0.310	0.485	300	0.175	600	0.310	0.485
	SGNS	400	0.295	400	0.333	0.627	400	0.295	400	0.333	0.627
	PMI-GSVD	800	0.267	600	0.318	0.585	5000	0.256	3000	0.308	0.564
	ROOT-CA	800	0.325	500	**0.400**	0.725	9000	0.324	800	**0.374**	**0.698**
	ROOTROOT-CA	600	**0.340**	400	0.396	**0.735**	1000	**0.332**	4000	0.359	0.690
	ROOT-CCA	600	0.315	400	0.392	0.706	900	0.298	800	0.355	0.653
Turk	RAW-CA	400	0.549	100	0.562	1.111	400	0.592	10000	0.588	1.181
	PMI-SVD	100	0.656	50	0.652	1.308	300	**0.677**	500	0.661	1.338
	PPMI-SVD	50	0.668	50	0.671	1.339	50	**0.677**	50	0.683	**1.361**
	GloVe	600	0.502	200	0.540	1.042	600	0.502	200	0.540	1.042
	SGNS	200	0.651	300	0.650	1.302	200	0.651	300	0.650	1.302
	PMI-GSVD	900	0.495	200	0.506	1.000	5000	0.563	10000	0.584	1.147
	ROOT-CA	50	0.649	50	**0.695**	**1.344**	100	0.661	50	**0.684**	1.345
	ROOTROOT-CA	50	**0.669**	50	0.666	1.334	50	0.664	300	0.673	1.337
	ROOT-CCA	50	0.633	50	0.672	1.305	100	0.665	100	0.678	1.343
Rare	RAW-CA	600	0.396	500	0.450	0.846	900	0.411	3000	0.465	0.875
	PMI-SVD	100	0.476	700	0.480	0.957	300	0.471	5000	0.464	0.936
	PPMI-SVD	100	0.483	400	0.470	0.952	100	0.475	6000	0.469	0.944
	GloVe	400	0.181	600	0.379	0.560	400	0.181	600	0.379	0.560
	SGNS	600	0.456	200	**0.532**	**0.988**	600	0.456	200	**0.532**	0.988
	PMI-GSVD	400	0.451	500	0.418	0.869	900	0.431	600	0.429	0.860
	ROOT-CA	400	0.468	400	0.501	0.970	600	**0.479**	7000	0.526	**1.006**
	ROOTROOT-CA	100	**0.503**	500	0.476	0.978	100	0.475	4000	0.478	0.953
	ROOT-CCA	200	0.469	200	0.505	0.974	600	0.469	900	0.511	0.979
SimLex-999	RAW-CA	4000	0.219	2000	0.322	0.541	8000	0.243	7000	0.327	0.571
	PMI-SVD	700	0.310	900	0.409	0.719	3000	0.315	900	0.372	0.687
	PPMI-SVD	700	0.309	500	0.393	0.702	3000	0.308	500	0.368	0.676
	GloVe	500	0.148	500	0.255	0.403	500	0.148	500	0.255	0.403
	SGNS	600	0.306	400	0.376	0.682	600	0.306	400	0.376	0.682
	PMI-GSVD	900	0.272	4000	0.365	0.637	5000	0.271	3000	0.312	0.583
	ROOT-CA	2000	0.295	900	0.415	0.710	5000	0.309	2000	**0.395**	**0.704**
	ROOTROOT-CA	700	**0.321**	900	0.410	**0.731**	700	**0.317**	900	0.376	0.693
	ROOT-CCA	1000	0.294	1000	**0.421**	0.715	7000	0.303	2000	0.391	0.693
total	RAW-CA		1.965		2.092	4.057		2.111		2.178	4.290
	PMI-SVD		2.445		2.562	5.007		2.465		2.433	4.897
	PPMI-SVD		2.476		2.556	5.033		2.478		2.501	4.979
	GloVe		1.427		2.006	3.433		1.427		2.006	3.433
	SGNS		2.376		2.442	4.819		2.376		2.442	4.819
	PMI-GSVD		1.997		2.075	4.072		2.069		2.082	4.151
	ROOT-CA		2.405		**2.635**	5.039		2.462		**2.637**	**5.098**
	ROOTROOT-CA		**2.525**		2.582	**5.107**		**2.484**		2.515	4.999
	ROOT-CCA		2.393		2.617	5.011		2.417		2.555	4.972

Table 4: Text8: the number of extreme values

	LT f_1	GT f_3	total
PMI	4,335	27,984	32,319
PPMI	0	27,984	27,984
WPMI	1,038,236	345,995	1,384,231
TTEST	50,560	627,046	677,606
ROOT-TTEST	5,985	448,860	454,845
ROOTROOT-TTEST	4,942	396,437	401,379
STRATOS-TTEST	0	400,703	400,703

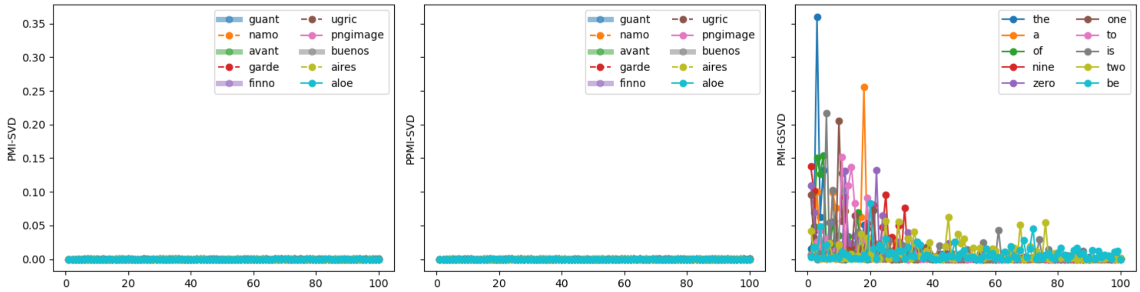


Figure 1: Text8: the contribution of the rows, corresponding to the top 10 extreme values, to the first 100 dimensions of PMI-SVD, PPMI-SVD, PMI-GSVD.

fined as values less than f_1 (LT f_1) or greater than f_3 (GT f_3). The first three rows in Table 4 show the number of extreme elements in the PMI, PPMI, WPMI matrices. The number of extreme values of the WPMI matrix (1,384,231) is much larger than that of PMI and PPMI (32,319 and 27,984). Furthermore, in WPMI the extremeness of values is much larger than in PMI and PPMI. Let the averaged contribution of each cell, expressed as a proportion, be $1/(11,815 \times 11,815)$. However, in WPMI, the most extreme entry, found for (the, the), contributes around 0.01126 to the total inertia. In PMI (PPMI) the most extreme entry is (guant, namo) or (namo, guant) and contributes around 3.1×10^{-6} (3.2×10^{-6}) to the total inertia. Figure 1 shows the contribution of the rows for the corresponding to top 10 extreme values, to the first 100 dimensions of PMI-SVD, PPMI-SVD, PMI-GSVD. The rows, corresponding to the top extreme values in the WPMI matrix, take up a much bigger contribution to the first dimensions of PMI-GSVD. For example, in PMI-GSVD, the "the" row contributes more than 0.3 to the third dimension, while in PMI-SVD and PPMI-SVD, the contributions are much more even. Thus the PMI-GSVD solution is hampered by extreme cells in the WPMI matrix that is decomposed. Similar results can be found for BNC in Supplementary materials C.

8.3 The results for three variants of CA

Now we compare the three variants of CA (ROOT-CA, ROOTROOT-CA, ROOT-CCA) with CA-RAW and the winner of the PMI-based methods, PPMI-SVD.

First, in Table 3, the three variants of CA perform much better than RAW-CA in each word similarity dataset and each corpus, both for $p = 0$ and $p = 0.5$. In the block at the

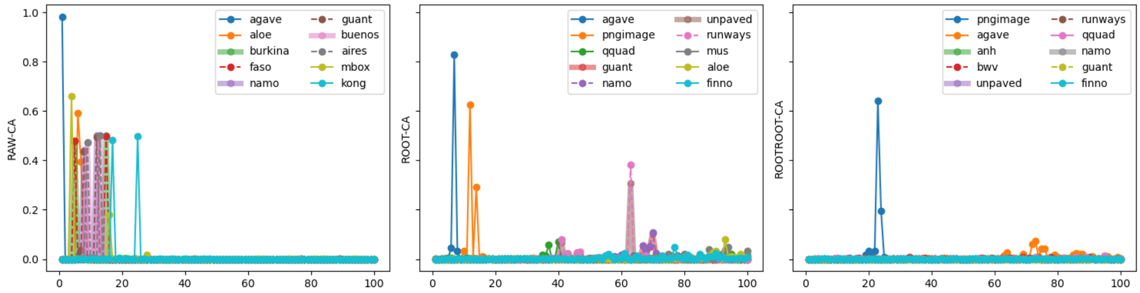


Figure 2: Text8: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of RAW-CA, ROOT-CA, ROOTROOT-CA.

bottom of Table 3, overall, the performance of the three variants is similar, where ROOT-CA outperforms ROOT-CCA slightly.

The lower part of Table 4 shows the number of extreme values of TTEST, ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST matrices for Text8. Similar with PMI, PPMI, WPMI, 94.2% of entries are ignored. The number of extreme values of the TTEST matrix (677,606) is larger than that of ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST (454,845, 401,379, and 400,703). Furthermore, in TTEST the extremeness of the extreme values is larger than those in ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST. For example, in TTEST the most extreme entry (agave, agave) contributes around 0.02117 to the total inertia, while in ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST, the most extreme entries (agave, agave), (pngimage, pngimage), and (agave, agave) contribute around 0.00325, 0.00119, and 0.00017, respectively. Figure 2 shows the contribution of the rows for the top 10 extreme values, to the first 100 dimensions of RAW-CA, ROOT-CA, and ROOTROOT-CA (The corresponding plot about ROOT-CCA is in Supplementary materials D). In RAW-CA, the rows, corresponding to top extreme values of TTEST, take up a big contribution to the first dimensions of RAW-CA. For example, in RAW-CA, the "agave" row contributes around 0.983 to the first dimension, while in ROOT-CA and ROOTROOT-CA, the contributions are much smaller which also holds for ROOT-CCA. Similar results can be found for BNC in Supplementary materials E. Thus, we infer that the extreme values in TTEST are the important reason that RAW-CA performs badly.

Second, in the rows of the block at the bottom of Table 3, the overall performances of ROOT-CA, ROOTROOT-CA, ROOT-CCA are comparable to or sometimes slightly better than PPMI-SVD. Specifically, ROOTROOT-CA and ROOT-CA achieve the highest ρ for Text8 and BNC corpora, respectively. Based on these results, no matter what we know about the corpus, ROOTROOT-CA and ROOT-CA appear to have potential to improve the performance in NLP tasks.

9 Conclusion and discussion

PMI is an important concept in natural language processing. In this paper, we theoretically compare CA with three PMI-based methods with respect to their objective functions.

CA is a weighted factorization of a matrix where the fitting function is $(p_{ij}/(p_{i+}p_{+j})-1)$ and the weighting function is the product of row margins and column margins $p_{i+}p_{+j}$. When the elements in the fitting function $(p_{ij}/(p_{i+}p_{+j})-1)$ of CA are small, CA is close to a weighted factorization of the PMI matrix where the weighting function is the product $p_{i+}p_{+j}$. This is because $(p_{ij}/(p_{i+}p_{+j})-1)$ is close to $\log(p_{ij}/(p_{i+}p_{+j}))$ when $(p_{ij}/(p_{i+}p_{+j})-1)$ is small.

The extracted word-context matrices are prone to overdispersion. To remedy the overdispersion, we presented ROOTROOT-CA. That is, we perform CA on the root-root transformation of the word-context matrix. We also apply CA to the square-root transformation of the word-context matrix (ROOT-CA). In addition, we present ROOT-CCA, described in Stratos et al. (2015), which is similar with ROOT-CA. The empirical comparison on word similarity tasks shows that ROOTROOT-CA achieves the best overall results in the Text8 corpus, and ROOT-CA achieves the best overall results in the BNC corpus. Overall, the performance of ROOT-CA and ROOTROOT-CA is slightly better than the performance of PMI-based methods.

Concluding, our theoretical and empirical comparisons about CA and PMI-based methods shed new light on SVD-based and PMI-based methods. Our results show that, regularly, in NLP tasks the performance can be improved by making use of ROOT-CA and ROOTROOT-CA.

In this paper, we explore ROOT-CA and ROOTROOT-CA, where ROOT-CA uses a power of 0.5 of the original elements x_{ij} while ROOTROOT-CA uses 0.25 of the original elements x_{ij}. Our aim was to study the performance of CA w.r.t. other methods, and for this purpose, focusing on values 0.25 and 0.5 was sufficient. It may be of interest to study a general power transformation x_{ij}^{δ} (or other power versions such as $(p_{ij}/(p_{i+}p_{+j}))^{\delta}$) where δ could range between any two non-negative values (Beh & Lombardo, 2024; Beh, Lombardo, & Wang, 2023; Cuadras & Cuadras, 2006; Greenacre, 2009). Here 0.5 and 0.25 are special cases of this general transformation.

Data availability

The Text8 corpus and BNC corpus that support the findings of this study are openly available by *Text8 dataset* (2006) and BNC Consortium (2007), respectively.

The five word similarity datasets for word similarity tasks are from `https://github.com/valentinp72/svd2vec/tree/master/svd2vec/datasets/similarities`

Acknowledgement

Author Qianqian Qi is supported by the China Scholarship Council.

References

Agresti, A. (2007). *An introduction to categorical data analysis*. Wiley. doi: 10.1002/0470114754

Alqahtani, Y., Al-Twairesh, N., & Alsanad, A. (2023). Improving sentiment domain adaptation for arabic using an unsupervised self-labeling framework. *Information Processing & Management*, *60*(3), 103338. doi: 10.1016/j.ipm.2023.103338

Bae, Y. S., Kim, K. H., Kim, H. K., Choi, S. W., Ko, T., Seo, H. H., . . . Jeon, H. (2021). Keyword extraction algorithm for classifying smoking status from unstructured bilingual electronic health records based on natural language processing. *Applied Sciences*, *11*(19), 8812. doi: 10.3390/app11198812

Bartlett, M. S. (1936). The square root transformation in analysis of variance. *Supplement to the Journal of the Royal Statistical Society*, *3*(1), 68–78. doi: 10.2307/2983678

Beh, E. J., & Lombardo, R. (2021). *An introduction to correspondence analysis*. Wiley. doi: 10.1002/9781119044482

Beh, E. J., & Lombardo, R. (2024). Correspondence analysis using the cressie–read family of divergence statistics. *International Statistical Review*, *92*(1), 17–42. doi: 10.1111/insr.12541

Beh, E. J., Lombardo, R., & Alberti, G. (2018). Correspondence analysis and the freeman–tukey statistic: A study of archaeological data. *Computational Statistics & Data Analysis*, *128*, 73–86. doi: 10.1016/j.csda.2018.06.012

Beh, E. J., Lombardo, R., & Wang, T.-W. (2023). Power transformations and reciprocal averaging. In E. J. Beh, R. Lombardo, & J. G. Clavel (Eds.), *Analysis of categorical data from historical perspectives: Essays in honour of shizuhiko nishisato* (pp. 173–199). Singapore: Springer Nature Singapore. doi: 10.1007/978-981-99-5329-5_11

Benzécri, J.-P. (1973). *L'analyse des données* (Vol. 1 and 2). Dunod Paris.

BNC Consortium. (2007). *British National Corpus, XML edition.* Retrieved from http://hdl.handle.net/20.500.14106/2554 (Literary and Linguistic Data Service)

Bruni, E., Boleda, G., Baroni, M., & Tran, N.-K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 136–145). Retrieved from https://aclanthology.org/P12-1015

Bullinaria, J. A., & Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, *39*, 510–526. doi: 10.3758/BF03193020

Cuadras, C. M., & Cuadras, D. (2006). A parametric approach to correspondence analysis. *Linear Algebra and its Applications*, *417*(1), 64-74. (Special Issue devoted to papers presented at the Workshop on Matrices and Statistics) doi: https://doi.org/10.1016/j.laa.2005.10.029

Curran, J. R. (2004). *From distributional to semantic similarity* (Doctoral dissertation, University of Edinburgh). Retrieved from http://hdl.handle.net/1842/563

Curran, J. R., & Moens, M. (2002). Improvements in automatic thesaurus extraction.

In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9* (p. 59–66). doi: 10.3115/1118627.1118635

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391–407. doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

Dodge, Y. (2008). *The concise encyclopedia of statistics*. Springer New York. doi: 10.1007/978-0-387-32833-1

Downing, J. (1981). How well does the 4th-root transformation work-reply. *Canadian Journal of Fisheries and Aquatic Sciences*, *38*(1), 127–129.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, *12*, 2121–2159. Retrieved from http://jmlr.org/papers/v12/duchi11a.html

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., & Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 281–285). doi: 10.1145/57167.57214

Egleston, B. L., Bai, T., Bleicher, R. J., Taylor, S. J., Lutz, M. H., & Vucetic, S. (2021). Statistical inference for natural language processing algorithms with a demonstration using type 2 diabetes prediction from electronic health record notes. *Biometrics*, *77*(3), 1089–1100. doi: 10.1111/biom.13338

Field, J., Clarke, K., & Warwick, R. M. (1982). A practical strategy for analysing multispecies distribution patterns. *Marine ecology progress series*, *8*(1), 37–52. Retrieved from http://www.jstor.com/stable/24814621

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. doi: 10.1145/503104.503110

France, R. L., & Heung, B. (2023). Density variability of COVID-19 face mask litter: A cautionary tale for pandemic PPE waste monitoring. *Journal of Hazardous Materials Advances*, *9*, 100220. doi: 10.1016/j.hazadv.2022.100220

Goodman, L. A. (1996). A single general method for the analysis of cross-classified data: Reconciliation and synthesis of some methods of pearson, yule, and fisher, and also some methods of correspondence analysis and association analysis. *Journal of the American Statistical Association*, *91*(433), 408–428. doi: 10.1080/01621459.1996.10476702

Greenacre, M. (1984). *Theory and applications of correspondence analysis*. Academic Press.

Greenacre, M. (2009). Power transformations in correspondence analysis. *Computational Statistics & Data Analysis*, *53*(8), 3107–3116. doi: 10.1016/j.csda.2008.09.001

Greenacre, M. (2010). Log-ratio analysis is a limiting case of correspondence analysis. *Mathematical Geosciences*, *42*, 129–134. doi: 10.1007/s11004-008-9212-2

Greenacre, M. (2017). *Correspondence analysis in practice*. CRC press. doi:

10.1201/9781315369983

Greenacre, M., & Hastie, T. (1987). The geometric interpretation of correspondence analysis. *Journal of the American statistical association*, *82*(398), 437–447. doi: 10.1080/01621459.1987.10478446

Guo, S., & Yao, N. (2021). Document vector extension for documents classification. *IEEE Transactions on Knowledge & Data Engineering*, *33*(8), 3062–3074. doi: 10.1109/TKDE.2019.2961343

Hill, F., Reichart, R., & Korhonen, A. (2015). SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, *41*(4), 665–695. doi: 10.1162/COLI_a_00237

Hollander, M., Wolfe, D. A., & Chicken, E. (2013). *Nonparametric statistical methods*. John Wiley & Sons.

Hsu, L. L., & Culhane, A. C. (2023). Correspondence analysis for dimension reduction, batch integration, and visualization of single-cell RNA-seq data. *Scientific Reports*, *13*, 1197. doi: 10.1038/s41598-022-26434-1

Jiang, C., Yu, H.-F., Hsieh, C.-J., & Chang, K.-W. (2018). Learning word embeddings for low-resource languages by PU learning. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 1024–1034). doi: 10.18653/v1/N18-1093

Jurafsky, D., & Martin, J. H. (2023). Vector semantics and embeddings. *Speech and language processing*, 1–34. Retrieved December 21, 2023, from https://web.stanford.edu/~jurafsky/slp3/6.pdf.

Kostensalo, J., Lidauer, M., Aernouts, B., Mäntysaari, P., Kokkonen, T., Lidauer, P., & Mehtiö, T. (2023). Short communication: Predicting blood plasma non-esterified fatty acid and beta-hydroxybutyrate concentrations from cow milk—addressing systematic issues in modelling. *animal*, *17*(9), 100912. doi: 10.1016/j.animal.2023.100912

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems* (Vol. 27). Retrieved from https://proceedings.neurips.cc/paper_files/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf

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. doi: 10.1162/tacl_a_00134

Luong, T., Socher, R., & Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the seventeenth conference on computational natural language learning* (pp. 104–113). Retrieved from https://aclanthology.org/W13-3512

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the international conference on Learning Representations ICLR'13.*

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (Vol. 26). Retrieved from https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

Nishisato, S., Beh, E. J., Lombardo, R., & Clavel, J. G. (2021). On the analysis of over-dispersed categorical data. In *Modern quantification theory: Joint graphical display, biplots, and alternatives* (pp. 215–231). Springer Singapore. doi: 10.1007/978-981-16-2470-4_11

Pakzad, A., & Analoui, M. (2021). A word selection method for producing interpretable distributional semantic word vectors. *Journal of Artificial Intelligence Research*, *72*, 1281–1305. doi: 10.1613/jair.1.13353

Peng, J., & Feldman, A. (2017). Automatic idiom recognition with word embeddings. In *Information management and big data* (pp. 17–29). Cham: Springer International Publishing. doi: 10.1007/978-3-319-55209-5_2

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). doi: 10.3115/v1/D14-1162

Podkorytov, M., Biś, D., Cai, J., Amirizirtol, K., & Liu, X. (2020). Effects of Architecture and Training on Embedding Geometry and Feature Discriminability in BERT. In *2020 International Joint Conference on Neural Networks (IJCNN)* (p. 1-8). doi: 10.1109/IJCNN48605.2020.9206645

Qi, Q., Hessen, D. J., Deoskar, T., & Van der Heijden, P. G. M. (2023). A comparison of latent semantic analysis and correspondence analysis of document-term matrices. *Natural Language Engineering*, 1–31. doi: 10.1017/S1351324923000244

Qi, Q., Hessen, D. J., & Van der Heijden, P. G. M. (2023). Improving information retrieval through correspondence analysis instead of latent semantic analysis. *Journal of Intelligent Information Systems*. doi: 10.1007/s10844-023-00815-y

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* (p. 337–346). doi: 10.1145/1963405.1963455

Raphael, E. B. (2023). Gendered representations in language: A corpus-based comparative study of adjective-noun collocations for marital relationships. *Theory and Practice in Language Studies*, *13*(5), 1191–1196. doi: 10.17507/tpls.1305.12

Roesler, O., Aly, A., Taniguchi, T., & Hayashi, Y. (2019). Evaluation of Word Representations in Grounding Natural Language Instructions Through Computational Human-Robot Interaction. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (p. 307-316). doi: 10.1109/HRI.2019.8673121

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536. doi: 10.1038/323533a0

Salle, A., & Villavicencio, A. (2023). Understanding the effects of negative (and positive) pointwise mutual information on word vectors. *Journal of Experimental & Theoretical Artificial Intelligence*, *35*(8), 1161-1199. doi: 10.1080/0952813X.2022.2072004

Salle, A., Villavicencio, A., & Idiart, M. (2016). Matrix factorization using window sampling and negative sampling for improved word representations. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 419–424). doi: 10.18653/v1/P16-2068

Samuel, D., Kutuzov, A., Øvrelid, L., & Velldal, E. (2023). Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023* (pp. 1954–1974). doi: 10.18653/v1/2023.findings-eacl.146

Sasaki, S., Heinzerling, B., Suzuki, J., & Inui, K. (2023). Examining the effect of whitening on static and contextualized word embeddings. *Information Processing & Management*, *60*(3), 103272. doi: 10.1016/j.ipm.2023.103272

Schwertman, N. C., Owens, M. A., & Adnan, R. (2004). A simple more general boxplot method for identifying outliers. *Computational Statistics & Data Analysis*, *47*(1), 165-174. doi: 10.1016/j.csda.2003.10.012

Shazeer, N., Doherty, R., Evans, C., & Waterson, C. (2016). Swivel: Improving embeddings by noticing what's missing. *arXiv preprint arXiv:1602.02215*.

Shi, T., & Liu, Z. (2014). Linking glove with word2vec. *arXiv preprint arXiv:1411.5595*.

Stratos, K., Collins, M., & Hsu, D. (2015). Model-based word embeddings from decompositions of count matrices. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (pp. 1282–1291). doi: 10.3115/v1/P15-1124

Text8 dataset. (2006). Retrieved October 3, 2023, from `http://mattmahoney.net/dc/text8.zip`.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley, Reading, MA.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*, 141–188. doi: 10.1613/jair.2934

Van der Heijden, P. G. M., De Falguerolles, A., & De Leeuw, J. (1989). A combined approach to contingency table analysis using correspondence analysis and loglinear analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *38*(2), 249–292. doi: 10.2307/2348058

Xin, X., Yuan, F., He, X., & Jose, J. M. (2018). Batch IS NOT heavy: Learning word representations from all samples. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1853–1862). doi: 10.18653/v1/P18-1172

Zhang, M., Palade, V., Wang, Y., & Ji, Z. (2022). Word representation using refined contexts. *Applied Intelligence*, *52*, 12347–12368. doi: 10.1007/s10489-021-02898-y

Supplementary A: An alternative coordinates system for CA

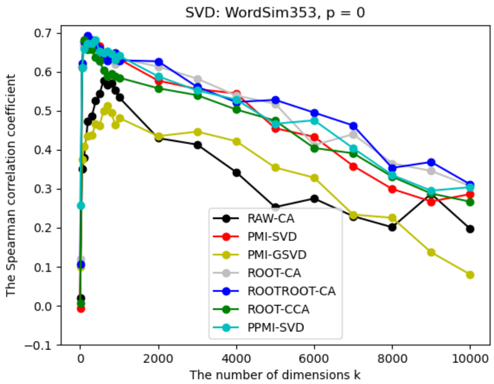
For row points i and i', with coordinates $\sigma_k \phi_{ik}$ and $\sigma_k \phi_{i'k}$ on dimension k in $K-$dimensional space we have

$$\begin{aligned}
\text{cosine}(\text{row}_i, \text{row}_{i'}) &= \frac{\sum_{k=1}^{K} (\phi_{ik}\sigma_k)(\phi_{i'k}\sigma_k)}{\sqrt{\sum_{k=1}^{K} (\phi_{ik}\sigma_k)^2 \cdot \sum_{k=1}^{K} (\phi_{i'k}\sigma_k)^2}} \\
&= \frac{\sum_{k=1}^{K} \left(p_{i+}^{-\frac{1}{2}} u_{ik}\sigma_k\right)\left(p_{i'+}^{-\frac{1}{2}} u_{i'k}\sigma_k\right)}{\sqrt{\sum_{k=1}^{K} \left(p_{i+}^{-\frac{1}{2}} u_{ik}\sigma_k\right)^2 \cdot \sum_{k=1}^{K} \left(p_{i'+}^{-\frac{1}{2}} u_{i'k}\sigma_k\right)^2}} \\
&= \frac{\sum_{k=1}^{K} (u_{ik}\sigma_k)(u_{i'k}\sigma_k)}{\sqrt{\sum_{k=1}^{K} (u_{ik}\sigma_k)^2 \cdot \sum_{k=1}^{K} (u_{i'k}\sigma_k)^2}},
\end{aligned} \tag{1}$$

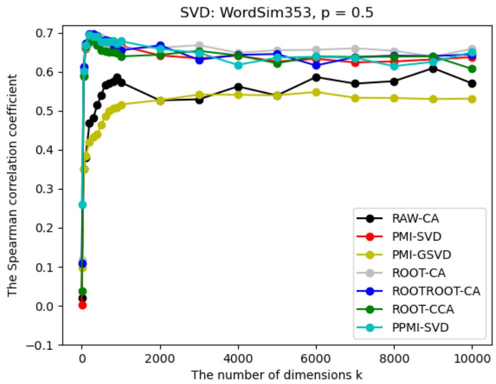
so the terms $p_{i+}^{-\frac{1}{2}}$ drop out of the equation. A similar result is found for column points.

Supplementary B: Plots for ρ as a function of k for SVD-based methods

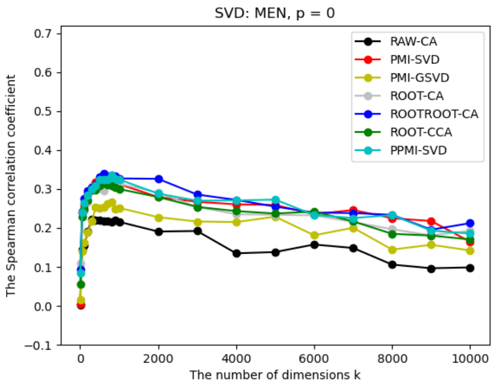
Plots are for ρ as a function of k for SVD-based methods.



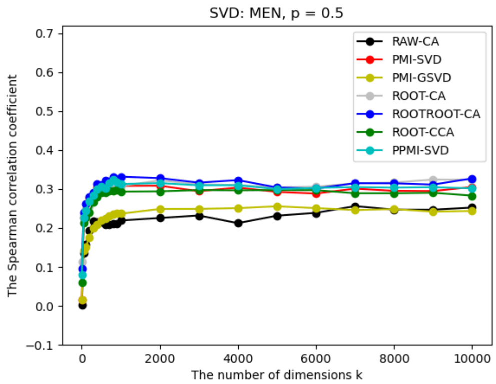
(a) WordSim353: $p = 0$



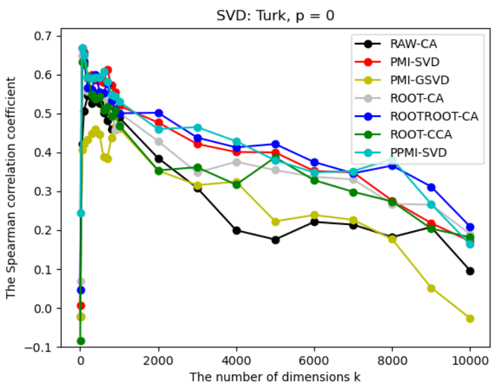
(b) WordSim353:$p = 0.5$



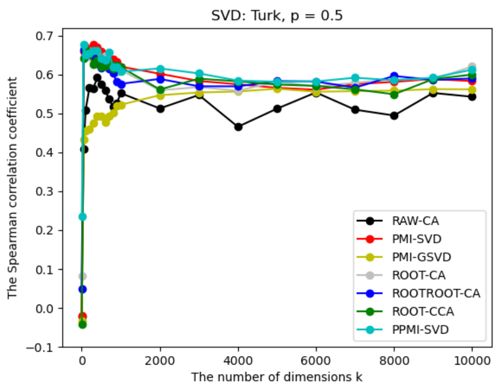
(c) MEN: $p = 0$



(d) MEN: $p = 0.5$

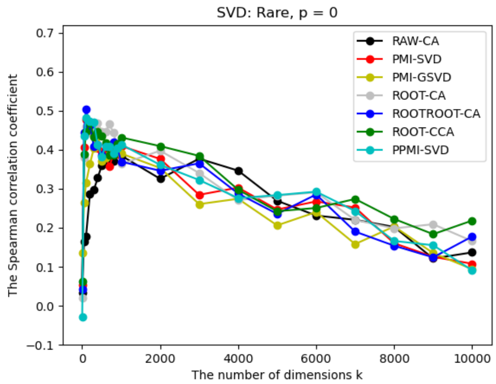


(e) Turk: $p = 0$

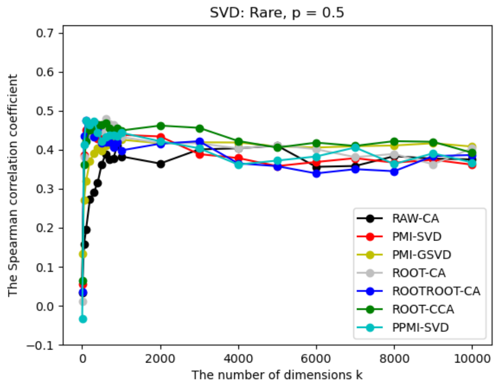


(f) Turk: $p = 0.5$

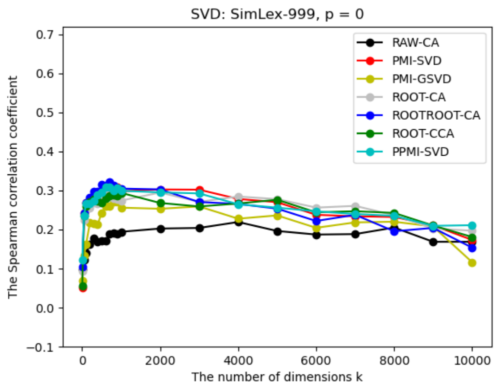
Figure B.1: Text8



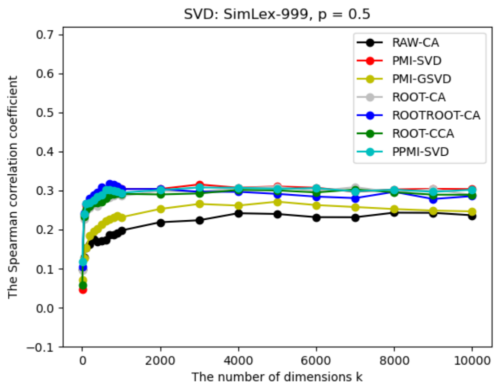
(g) Rare: $p = 0$



(h) Rare: $p = 0.5$

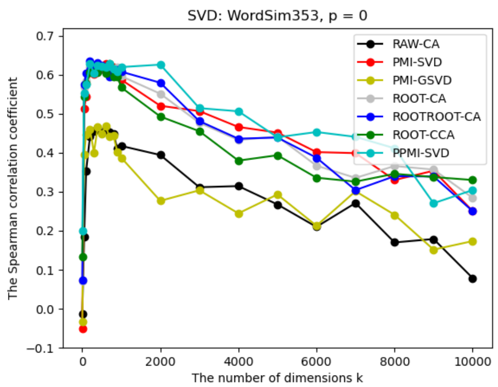


(i) SimLex-999: $p = 0$

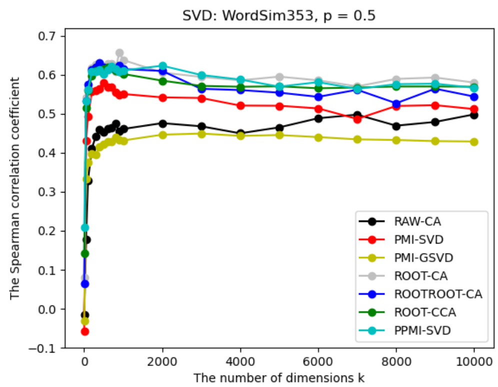


(j) SimLex-999: $p = 0.5$

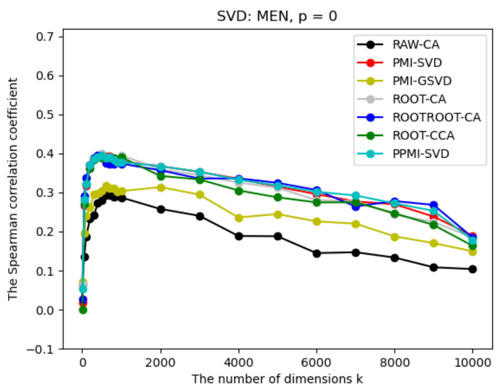
Figure B.1: Text8



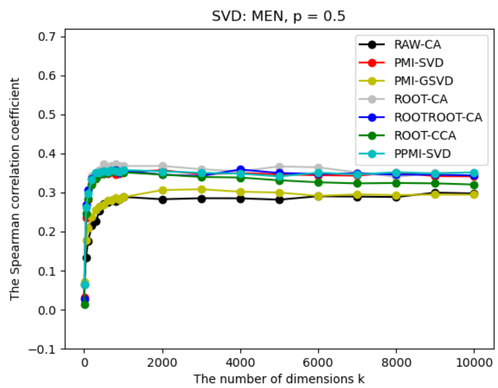
(a) WordSim353: $p = 0$



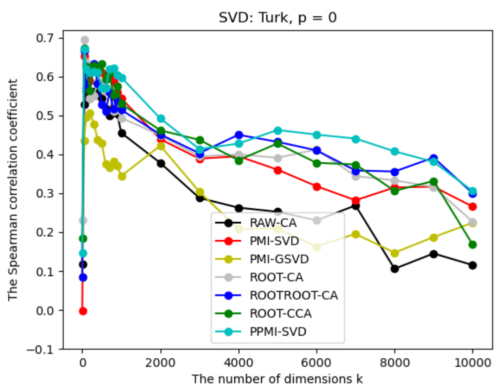
(b) WordSim353: $p = 0.5$



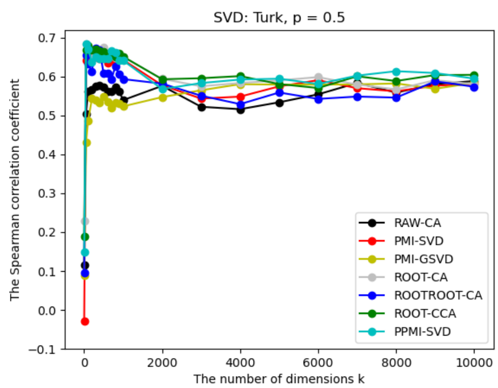
(c) MEN: $p = 0$



(d) MEN: $p = 0.5$

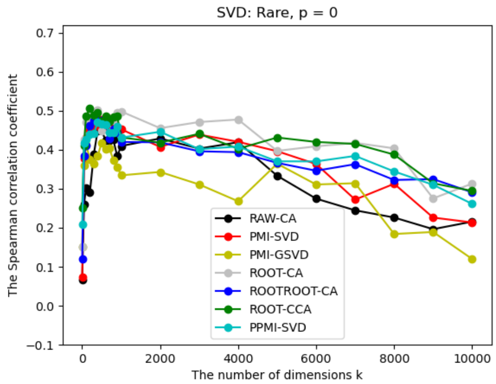


(e) Turk: $p = 0$

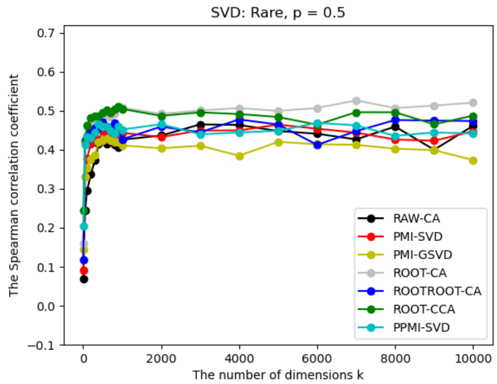


(f) Turk: $p = 0.5$

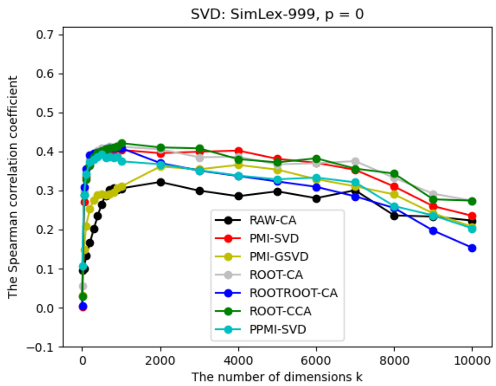
Figure B.2: BNC



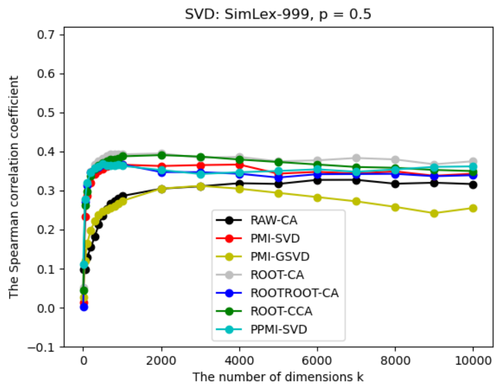
(g) Rare: $p = 0$



(h) Rare: $p = 0.5$



(i) SimLex-999: $p = 0$



(j) SimLex-999: $p = 0.5$

Figure B.2: BNC

Supplementary C: BNC: the number and sizes of extreme values of PMI, PPMI, and WPMI, and plots showing the contribution of the rows about PMI-SVD, PPMI-SVD, and PMI-GSVD

Table C.1, part PMI, PPMI, WPMI, shows the number of extreme values of PMI, PPMI, WPMI matrices. We only include non-zero pairs of PMI matrix because the PMI matrix is sparse: 84.1% of the entries are zero. The corresponding 84.1% of entries in PPMI and WPMI are also ignored. The number of extreme values of WPMI matrix (2,525,345) is much larger than that of PMI and PPMI (141,366 and 405,830). Furthermore, in WPMI the extremeness of the extreme values is much larger than those in PMI and PPMI. For example, where the average contribution of each cell is $1/(11,332 \times 11,332)$, in WPMI the most extreme entry (the, the) contributes around 0.01150 to the total inertia, while in PMI (PPMI), the most extreme entry (ee, ee) contributes around 2.2×10^{-6} (2.7×10^{-6}) to the total inertia. Figure C.1 shows the contribution of the rows, corresponding to top 10 extreme values, to the first 100 dimensions of PMI-SVD, PPMI-SVD, PMI-GSVD. The rows, corresponding to the top extreme values of WPMI, take up a bigger contribution to the first dimensions of PMI-GSVD. For example, in PMI-GSVD, the “the” row contributes

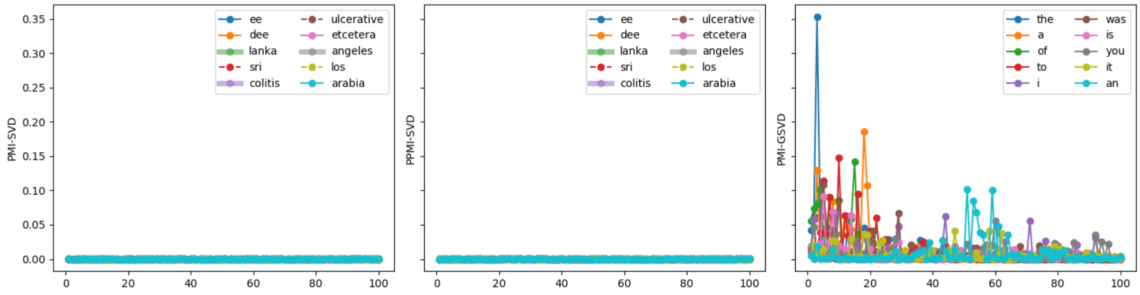


Figure C.1: BNC: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of PMI-SVD, PPMI-SVD, PMI-GSVD.

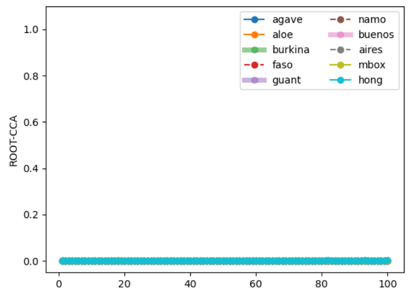


Figure D.1: Text8: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of ROOT-CCA.

more than 0.3 to the third dimension, while in PMI-SVD and PPMI-SVD, the contributions are much smaller.

Table C.1: BNC: the number of extreme values

	LT f_1	GT f_3	total
PMI	13,982	127,384	141,366
PPMI	0	405,830	405,830
WPMI	2,037,800	487,545	2,525,345
TTEST	334,512	1,480,336	1,814,848
ROOT-TTEST	35,418	927,470	962,888
ROOTROOT-TTEST	31,234	750,433	781,667
STRATOS-TTEST	0	1,173,717	1,173,717

Supplementary D: Text8: plots showing the contribution of the rows about ROOT-CCA

Figure D.1 shows the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of ROOT-CCA

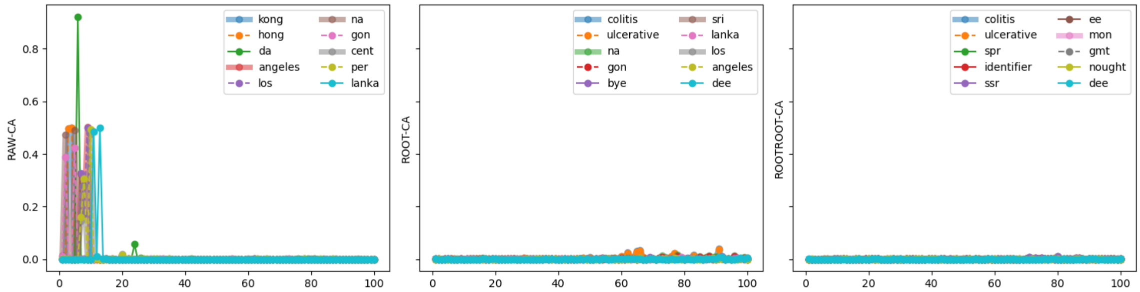


Figure E.1: BNC: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of RAW-CA, ROOT-CA, ROOTROOT-CA.

Supplementary E: BNC: the number and sizes of extreme values of TTEST, ROOT-TTEST, ROOTROOT-TTEST, and STRATOS-TTEST, and plots showing the contribution of the rows about RAW-CA, ROOT-CA, ROOTROOT-CA, and ROOT-CCA

The bottom part of Table C.1 shows the number of extreme values of TTEST, ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST matrices. Similar with PMI, PPMI, WPMI, 84.1% of entries are ignored. The number of extreme values of TTEST matrix (1,814,848) is much larger than that of ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST (962,888, 781,667, and 1,173,717). Furthermore, in TTEST the extremeness of the extreme values is much larger than in ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST. For example, in TTEST the most extreme entry (kong, hong) or (hong, kong) contributes around 0.00965 to the total inertia, while in ROOT-TTEST, ROOTROOT-TTEST and STRATOS-TTEST, the most extreme entries (colitis, ulcerative) or (ulcerative, colitis), (colitis, ulcerative) or (colitis, ulcerative), (hong, kong) or (kong, hong) contribute around 0.00047, 0.00003, and 0.00008 respectively. Figure E.1 shows the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of RAW-CA, ROOT-CA, and ROOTROOT-CA. The corresponding plot about ROOT-CCA is in Figure E.2. In RAW-CA, the rows, corresponding to top extreme values of TTEST, have a big contribution to the first dimensions of RAW-CA, while in ROOT-CA and ROOTROOT-CA, the contributions are much smaller, which also holds for ROOT-CCA.

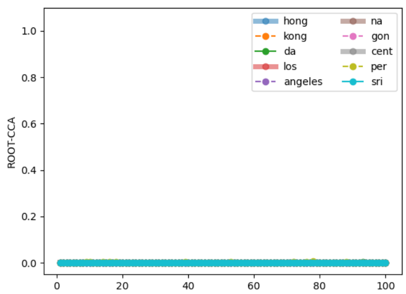


Figure E.2: BNC: the contribution of the rows, corresponding to top 10 extreme values, to first 100 dimensions of ROOT-CCA.