Improving information retrieval through correspondence analysis instead of latent semantic analysis

Qianqian Qi[1*], David J. Hessen[1] and Peter G. M. van der Heijden[1,2]

[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.

*Corresponding author(s). E-mail(s): q.qi@uu.nl;
Contributing authors: d.j.hessen@uu.nl;
p.g.m.vanderheijden@uu.nl;

Abstract

The initial dimensions extracted by latent semantic analysis (LSA) of a document-term matrix have been shown to mainly display marginal effects, which are irrelevant for information retrieval. To improve the performance of LSA, usually the elements of the raw document-term matrix are weighted and the weighting exponent of singular values can be adjusted. An alternative information retrieval technique that ignores the marginal effects is correspondence analysis (CA). In this paper, the information retrieval performance of LSA and CA are empirically compared. Moreover, it is explored whether the two weightings also improve the performance of CA. The results for four empirical datasets show that CA always performs better than LSA. Weighting the elements of the raw data matrix can improve CA; however, it is data dependent and the improvement is small. Adjusting the singular value weighting exponent often improves the performance of CA; however, the extent of the improvement depends on the dataset and the number of dimensions.

Keywords: Singular value decomposition, Singular value weighting exponent, Initial dimensions, Information retrieval

1 Introduction

In information retrieval, the similarity between a given user query and each document in a document-term matrix is calculated and documents with high similarity are returned (Kolda and O'leary, 1998; Zhang et al, 2011; Al-Qahtani et al, 2015; Guo et al, 2022). Latent semantic analysis (LSA) has been used as a common baseline for information retrieval (Paralı et al, 2019; Duan et al, 2021; Chang et al, 2021). Compared to Word2Vec (Skip-Gram model) LSA showed a better performance in extracting relevant semantic patterns in dream reports (Altszyler et al, 2016). LSA also outperformed neural network methods (such as ELMo word embeddings) in text classification tasks for educational data (Phillips et al, 2021).

New methods that rely on LSA have been proposed (Azmi et al, 2019; Gupta and Patel, 2021; Hassani et al, 2021; Suleman and Korkontzelos, 2021; Horasan, 2022; Patil, 2022). For example, Gupta and Patel (2021) proposed an algorithm for text summarization that uses LSA, TF-IDF keyword extractor, and BERT encoder model. The algorithm performed better than latent Dirichlet allocation. Horasan (2022) proposed a collaborative filtering-based recommendation system using LSA and achieved good performance. Patil (2022) developed a new promising procedure for information retrieval using LSA and TF-IDF.

Weighting the elements of the raw document-term matrix is a common and effective method to improve the performance of LSA (Dumais, 1991; Horasan et al, 2019; Bacciu et al, 2019). LSA usually involves the SVD of a raw or preprocessed document-term matrix. In addition, Caron (2001) proposed changing the weighting exponent of the singular values in LSA to improve information retrieval. His results showed that adjusting the weighting exponent of singular values improves the performance of information retrieval. Since Caron (2001), singular value weighting exponents have been studied and applied in word embeddings generated from word-context matrices (Bullinaria and Levy, 2012; Österlund et al, 2015; Drozd et al, 2016; Yin and Shen, 2018). Other variants that change the singular value weighting exponent have been studied in word embeddings created by Word2Vec and GloVe (Mu and Viswanath, 2018; Liu et al, 2019).

The larger the weighting exponent of the singular values, the higher is the emphasis given to the initial dimensions. According to the experimental results of Caron (2001), giving more emphasis to initial dimensions can often improve the performance of information retrieval on standard test datasets, whereas giving more emphasis to initial dimensions can decrease the performance on question/answer matching. Papers about word embeddings tend to reduce the contribution of initial dimensions to improve performance (Bullinaria and Levy, 2012; Österlund et al, 2015; Drozd et al, 2016; Yin and Shen, 2018; Mu and Viswanath, 2018; Liu et al, 2019), although the optimal value of the singular value weighting exponent is task dependent (Österlund et al, 2015). Bullinaria and Levy (2012) reported that assigning less weight to initial dimensions leads to improved performance for TOEFL, distance comparison, semantic categorization, and clustering purity tasks on a word-context matrix

created from the ukWaC corpus (Baroni et al, 2009). They argued that the general pattern appears to be that the initial dimensions tend not to contribute the most useful information about semantics and have a large "noise" component that is best removed or reduced.

Capturing associations between documents and terms appears necessary for the success of LSA in computing science; however, the solution of LSA is a mix of the associations between documents and terms, and marginal effects arising from the lengths of documents and marginal frequencies of terms (Qi et al, 2023). Hu et al (2003) and Qi et al (2023) showed that margins play an important role in the first dimensions extracted by LSA.

Correspondence analysis (CA) is another information retrieval technique that uses SVD (Greenacre, 1984; Morin, 2004; Greenacre, 2017; Beh and Lombardo, 2021). In computing science, CA has not been explored as much as LSA. CA is usually used to make two-dimensional graphical displays (Hou and Huang, 2020; Arenas-Márquez et al, 2021; Van Dam et al, 2021). For example, Arenas-Márquez et al (2021) depicted a biplot using CA to show that the document encoding of convolutional neural encoder can emphasize the dissimilarity between documents belonging to different classes. Unlike LSA, CA ignores the information on marginal frequency differences between documents and between terms from the solution by preprocessing the data, and it only focuses on the relationships between documents and terms (Qi et al, 2023). Thus, CA seems more suitable for information retrieval.

Séguéla and Saporta (2011) and Qi et al (2023) experimentally compared LSA and CA for text clustering and text categorization, respectively, and they found that CA performed better than LSA. Although LSA was originally proposed for information retrieval, an empirical comparison between LSA and CA continues to remain lacking in this field. In this paper, therefore, three English datasets and one Dutch dataset are used to compare the performance of LSA and CA in information retrieval.

Whereas LSA owes its popularity to its applicability to different matrices, in CA, it is unusual to weight the elements of the raw document-term matrix. Processing the raw document-term matrix is an integral part of CA (Greenacre, 1984, 2017; Beh and Lombardo, 2021). CA is based on the SVD of the matrix of standardized residuals. Here, however, we study the CA of document-term matrices whose entries are weighted to see if this has an impact on the performance of CA. In addition, based on the success of adjusting the weighting exponent of singular values in LSA, we will explore whether this is also successful in CA.

In summary, this work makes three contributions. First, to compare LSA and CA in information retrieval. Second, to explore whether weightings, including the weighting of the elements of the raw document-term matrix and the adjusting of the singular value weighting exponent, can improve the performance of CA. Third, to study what the initial dimensions of LSA correspond to and whether CA is effective in ignoring the useless information in the raw or pre-processed document-term matrix that contributes a large part of the initial dimensions extracted by LSA. We extensively compare the performances of

LSA and CA applied to four datasets using Euclidean distance, dot similarity, and cosine similarity.

The paper is organized as follows. In Section 2, LSA and CA are described in brief. Section 3 presents the methodology used in this paper. The results for Euclidean distance are presented in Section 4, and the results for dot similarity and cosine similarity are presented in Section 5. Finally, Section 6 concludes and discusses the results.

2 LSA and CA

In this section, we briefly describe LSA and CA. We refer the readers to Qi et al (2023) for a more detailed presentation of the methods.

2.1 LSA

Consider a raw document-term matrix $\boldsymbol{F} = [f_{ij}]$ with m rows ($i = 1, ..., m$) and n columns ($j = 1, ..., n$), where the rows represent documents and the columns represent terms. Weighting might be used to prevent the differential lengths of documents from considerably affecting the representation, or to impose certain preconceptions about which terms are more important (Deerwester et al, 1990). The weighted element a_{ij} for term j in document i is

$$a_{ij} = L(i, j) \times G(j) \times N(i), \tag{1}$$

where the local weighting term $L(i, j)$ is the weight of term j in document i, $G(j)$ is the global weight of term j in the entire set of documents, and $N(i)$ is the weighting component for document i. The popular TF-IDF can be written in the form $L(i, j) = f_{ij}, G(j) = 1 + \log_2(n\text{docs}/df_j), N(i) = 1$, where $n$docs is the number of documents in the set and df_j is the number of documents where term j appears (Dumais, 1991). The SVD of $\boldsymbol{A} = [a_{ij}]$ is

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{2}$$

where $\boldsymbol{U}^T\boldsymbol{U} = \mathbf{I}$, $\boldsymbol{V}^T\boldsymbol{V} = \mathbf{I}$, and $\boldsymbol{\Sigma}$ is a diagonal matrix with singular values on the diagonal in the descending order. We denote matrices that contain the first k columns of $\boldsymbol{U}$, first k columns of $\boldsymbol{V}$, and k largest singular values of $\boldsymbol{\Sigma}$ by $\boldsymbol{U}_k$, $\boldsymbol{V}_k$, and $\boldsymbol{\Sigma}_k$, respectively. Then, $\boldsymbol{U}_k\boldsymbol{\Sigma}_k(\boldsymbol{V}_k)^T$ provides the optimal rank-k approximation of $\boldsymbol{A}$ in a least-squares sense, which shows that SVD can be used for data reduction. In LSA, the rows of $\boldsymbol{U}_k\boldsymbol{\Sigma}_k$ and $\boldsymbol{V}_k\boldsymbol{\Sigma}_k$ provide the coordinates of row and column points, respectively. Euclidean distances between the rows of $\boldsymbol{U}_k\boldsymbol{\Sigma}_k$ ($\boldsymbol{V}_k\boldsymbol{\Sigma}_k$) approximate those between the rows (columns) of $\boldsymbol{A}$.

Representing out-of-sample documents or queries in the k-dimensional subspace of LSA is important for many applications including information retrieval. Suppose that the new weighted document is a row vector $\boldsymbol{d}$. Since $\boldsymbol{V}^T\boldsymbol{V} = \mathbf{I}$ and $\boldsymbol{U}^T\boldsymbol{U} = \mathbf{I}$, we have

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{U}_k\boldsymbol{\Sigma}_k \tag{3}$$

and

$$\boldsymbol{A}^T\boldsymbol{U}_k = \boldsymbol{V}_k\boldsymbol{\Sigma}_k \tag{4}$$

Therefore, using Equation (3), the coordinates of the out-of-sample document $\boldsymbol{d}$ in the k-dimensional subspace of LSA is $\boldsymbol{dV}_k$. Similarly, using Equation (4), the coordinates of the out-of-sample term $\boldsymbol{t}$ (represented as row vector) in the k-dimensional subspace of LSA is $\boldsymbol{tU}_k$.

As in Qi et al (2023), we first use a small dataset to illustrate LSA. This small dataset is introduced in Aggarwal (2018) (see Table 1), and it contains 6 documents. For each document, we are interested in the frequency of occurrence of six terms. The first three documents primarily refer to cats, the last two primarily to cars, and the fourth to both. The fourth term, jaguar, is polysemous because it can refer to either a cat or a car.

Table 1: A document-term matrix $\boldsymbol{F}$: size 6×6

	lion	tiger	cheetah	jaguar	porsche	ferrari
doc1	2	2	1	2	0	0
doc2	2	3	3	3	0	0
doc3	1	1	1	1	0	0
doc4	2	2	2	3	1	1
doc5	0	0	0	1	1	1
doc6	0	0	0	2	1	2

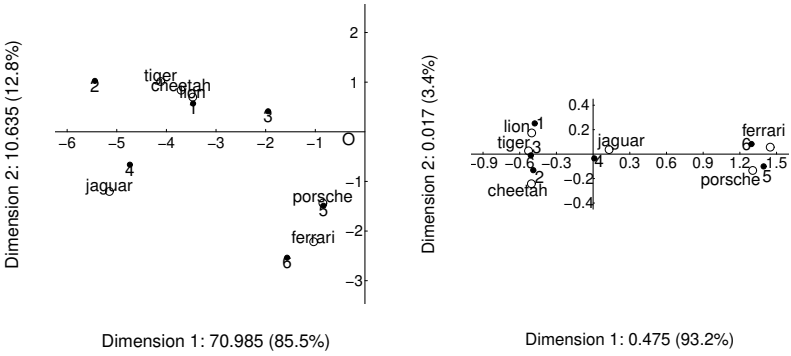
In the LSA of the raw document-term matrix (LSA-RAW), the rows and columns of $\boldsymbol{F}$ are not weighted, and therefore, we can replace $\mathbf{A}$ in Equation (2) by $\boldsymbol{F}$. The coordinates of the documents and of the terms for LSA-RAW in the first two dimensions are $\boldsymbol{U}_2\boldsymbol{\Sigma}_2$ and $\boldsymbol{V}_2\boldsymbol{\Sigma}_2$, respectively. Figure 1a shows the two-dimensional plot of the documents and terms. Cat terms (*lion*, *cheetah*, and *tiger*) are close together; car terms (*porsche* and *ferrari*) are close together; car documents (5 and 6) are close together. However, the cat documents (1, 2, and 3) are not close together, neither is document 4 in between cat documents and car documents, and neither is *jaguar* in between cat terms and car terms. This can be attributed to the fact that LSA displays both the relationships between documents and terms and the sizes of the documents and terms: for the latter, *jaguar*, for example, is used most often in the documents and is furthest away from the origin.

2.2 CA

In CA, an SVD is applied to the matrix of standarized residuals given by Greenacre (2017)

$$\boldsymbol{S} = \boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P} - \boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}} \tag{5}$$

where $\boldsymbol{P} = [p_{ij}]$ is the matrix of joint observed proportions with $p_{ij} = f_{ij}/\sum_i\sum_j f_{ij}$, $\boldsymbol{D}_r$ is a diagonal matrix with $r_i = \sum_j p_{ij}$ $(i = 1, 2, \cdots, m)$ on the diagonal, $\boldsymbol{D}_c$ is a diagonal matrix with $c_j = \sum_i p_{ij}$ $(j = 1, 2, \cdots, n)$ on the diagonal, and $\boldsymbol{E} = [r_i c_j]$ is the matrix of expected proportions under the statistical independence of the documents and the terms. The elements of



(a) LSA-RAW (b) CA

Fig. 1: A two-dimensional plot of documents and terms for (a) LSA-RAW, (b) CA (Qi et al, 2023).

$\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ are standardized residuals under the statistical independence model. The sum of squares of these elements yields the total inertia, i.e., the Pearson χ^2 statistic divided by sample size $\sum_i \sum_j f_{ij}$. By taking the SVD of the matrix of standardized residuals, we get

$$\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P}-\boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{6}$$

In CA, the rows of $\boldsymbol{\Phi}_k\boldsymbol{\Sigma}_k$ and $\boldsymbol{\Gamma}_k\boldsymbol{\Sigma}_k$ provide the coordinates of row and column points, respectively, where $\boldsymbol{\Phi}_k = \boldsymbol{D}_r^{-\frac{1}{2}}\boldsymbol{U}_k$ and $\boldsymbol{\Gamma}_k = \boldsymbol{D}_c^{-\frac{1}{2}}\boldsymbol{V}_k$. The weighted sum of the coordinates is 0: $\sum_i r_i\phi_{ik} = 0 = \sum_j c_j\gamma_{jk}$. Euclidean distances between the rows of $\boldsymbol{\Phi}_k\boldsymbol{\Sigma}_k$ ($\boldsymbol{\Gamma}_k\boldsymbol{\Sigma}_k$) approximate χ^2-distances between the rows (columns) of $\boldsymbol{F}$, where the squared χ^2-distance between rows k and l is

$$\delta_{kl}^2 = \sum_j \frac{(p_{kj}/r_k - p_{lj}/r_l)^2}{c_j} \tag{7}$$

In Equation (7), the rows are transformed into vectors of conditional proportions adding up to 1 for each row, such as the kth row: p_{kj}/r_k, $j = 1, 2, \cdots, n$, and the differences between the column elements for column j in the transformed rows are corrected for c_j, which represents the size of column j.

The transition formulas are

$$\boldsymbol{D}_r^{-1}\boldsymbol{P}\boldsymbol{\Gamma}_k = \boldsymbol{\Phi}_k\boldsymbol{\Sigma}_k \tag{8}$$

and

$$\boldsymbol{D}_c^{-1}\boldsymbol{P}^T\boldsymbol{\Phi}_k = \boldsymbol{\Gamma}_k\boldsymbol{\Sigma}_k \tag{9}$$

Equation (8) shows that the row points are in the weighted averages of the column points when rows of $\boldsymbol{D}_r^{-1}\boldsymbol{P}$ are used as weights, and Equation (9) shows that the column points are in the weighted averages of the row points simultaneously.

According to Equation (8), a new document $\boldsymbol{d}$, represented by a row vector, can be projected onto the k-dimensional subspace by placing it in the weighted average of the column points using $(\boldsymbol{d}/\sum_{j=1}^{n} d_j)\boldsymbol{\Gamma}_k$. This can be similarly done for a new term $\boldsymbol{t}$.

For the CA of Table 1, the coordinates of the documents and terms for CA in the first two dimensions are $\boldsymbol{\Phi}_2\boldsymbol{\Sigma}_2$ and $\boldsymbol{\Gamma}_2\boldsymbol{\Sigma}_2$, respectively. Figure 1b shows a two-dimensional plot of the documents and terms. Cat terms (*lion*, *cheetah*, and *tiger*) are close together; car terms (*porsche* and *ferrari*) are close together; *jaguar* is in between cat and car terms; car documents (5 and 6) are close together, cat documents (1, 2, and 3) are close together; and document 4 is in between cat and car documents. All data properties are found in Figure 1b. A comparison of Figures 1b and 1a suggests that CA provides a clearer visualization of the important aspects of the data than LSA. This is because the coordinates of each dimension are orthogonal to the margins due to $\sum_i r_i\phi_{ik} = 0 = \sum_j c_j\gamma_{jk}$, and CA focuses only on the relationship between the documents and the terms.

3 Methodology

In this section, we introduce the CA of a document-term matrix whose entries are weighted. We also discuss how the influence of the initial dimensions can be studied. Subsequently, we describe the study design, datasets, and evaluation methods used.

3.1 CA of a document-term matrix of weighted frequencies

Weighting the entries of the raw document-term matrix is an effective method for improving the performance of LSA, and this motivates us to study the weighting of the elements of the input matrix of CA. So, we try to improve the performance of CA by using the same weighting methods as in LSA.

The processing of the raw data matrix by $\boldsymbol{D}_r^{-\frac{1}{2}}(\boldsymbol{P} - \boldsymbol{E})\boldsymbol{D}_c^{-\frac{1}{2}}$ (see Equation (5)) is considered an integral part of CA. This processing step effectively eliminates the margins, which allows CA to focus on the relationships between documents and terms. The weighting of the entries of the raw document-term matrix in Equation (1), such as by TF-IDF, can be used to assign higher values to terms with more indicative of the meaning of documents. Thus, the weighting of the entries of the raw document-term matrix may also be an effective method for improving the performance of CA.

To perform the CA of a document-term matrix of weighted frequencies, we first use Equation (1) to obtain a document-term matrix $\boldsymbol{A}$ of weighted frequencies, and then, we perform CA on this matrix $\boldsymbol{A}$ instead of $\boldsymbol{F}$.

3.2 Changing the contributions of the initial dimensions in SVD

Caron (2001) proposed adjusting the relative strengths of vector components in LSA using $\boldsymbol{U}_k\boldsymbol{\Sigma}_k^{\alpha}$ or $\boldsymbol{V}_k\boldsymbol{\Sigma}_k^{\alpha}$ as coordinates instead of $\boldsymbol{U}_k\boldsymbol{\Sigma}_k$ or $\boldsymbol{V}_k\boldsymbol{\Sigma}_k$, where α is the singular value weighting exponent that adjusts the importance of the dimensions. The weighting exponent α determines how components are weighted relative to the standard $\alpha = 1$ case described in Section 2.1. In comparison to $\alpha = 1$, $\alpha < 1$ gives less emphasis to initial dimensions, and $\alpha > 1$, more emphasis.

Bullinaria and Levy (2012) used both weighting exponent $\alpha < 1$ and the exclusion of initial dimensions, which led to performance improvements of a similar degree. They argued that the general pattern appears to be that the dimensions with the highest singular values tend not to contribute the most useful information about semantics and have a large "noise" component that is best removed or reduced. However, it is unclear what the initial dimensions actually correspond to. Given this context, we change the contributions of the initial dimensions extracted by both LSA and CA and compare their performances. We explore whether the performance of CA can be improved by adjusting the singular value weighting exponent using $\boldsymbol{\Phi}_k\boldsymbol{\Sigma}_k^{\alpha}$ or $\boldsymbol{\Gamma}_k\boldsymbol{\Sigma}_k^{\alpha}$ as coordinates instead of $\boldsymbol{\Phi}_k\boldsymbol{\Sigma}_k$ or $\boldsymbol{\Gamma}_k\boldsymbol{\Sigma}_k$. That is, we try to improve the performance of CA by using the method (adjusting the singular weighting exponent) used in LSA.

We use Table 1 to illustrate the impact of α on singular values and coordinates. We use $\alpha = 0.5$, $\alpha = 1$, and $\alpha = 1.5$. In the literature, we regularly encounter $\alpha = 0.5$ because it relates to

$$\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \left(\boldsymbol{U}\boldsymbol{\Sigma}^{1/2}\right)\left(\boldsymbol{\Sigma}^{1/2}\boldsymbol{V}^T\right) \tag{10}$$

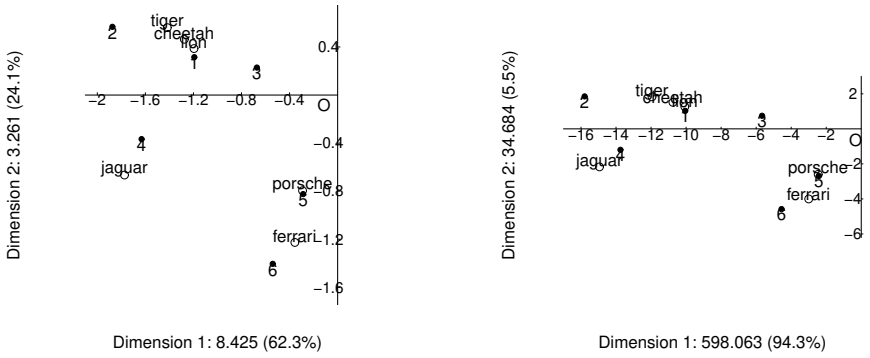
which can then be used for making biplots (Gabriel, 1971) using coordinate pairs $\boldsymbol{U}_2\boldsymbol{\Sigma}_2^{1/2}$ and $\boldsymbol{V}_2\boldsymbol{\Sigma}_2^{1/2}$. In practice, one often sees the use of the coordinate pair $\boldsymbol{U}_2\boldsymbol{\Sigma}_2$ and $\boldsymbol{V}_2\boldsymbol{\Sigma}_2$; however, this is not a biplot representation as $\boldsymbol{\Sigma}_2$ is used twice. In a biplot, if the row points are $\boldsymbol{U}_2\boldsymbol{\Sigma}_2^{a}$, then the column points are $\boldsymbol{V}_2\boldsymbol{\Sigma}_2^{1-a}$, i.e., any entry of the matrix is approximated by the inner product of the corresponding row and column vectors. Hereafter, we do not make a biplot; instead, we make a symmetric plot where documents and terms have the same value of α because symmetric coordinates are usually used in experiments (Dumais et al, 1988; Deerwester et al, 1990; Berry et al, 1995; Levy et al, 2015).

Table 2 lists the singular values to the power α: σ^{α}, the squared singular values to the power α: $\sigma^{2\alpha}$, and proportions $\sigma^{2\alpha}/\sum_{\sigma}\sigma^{2\alpha}$, where we refer to the total sum of squared singular values to the power of α, $\sum_{\sigma}\sigma^{2\alpha}$, as α–inertia. These proportions show how the sum of the Euclidean distances of all components to the origin is distributed over the components. The greater α is, the more emphasis is given to the initial components and less emphasis to the latter ones. The first dimension accounts for 0.623, 0.855, and 0.943 of α-inertia, while the fifth dimension accounts for 0.020, 0.001, and 0.000, with α being 0.5, 1, and 1.5, respectively. The standard LSA solution has $\alpha = 1$.

Table 2: The σ^{α}, $\sigma^{2\alpha}$, and the proportion of explained α-inertia $\sigma^{2\alpha} / \sum_{\sigma} \sigma^{2\alpha}$ for each dimension of LSA-RAW.

	dim1	dim2	dim3	dim4	dim5
$\sigma^{0.5}$	2.903	1.806	0.994	0.758	0.522
σ^{1}	8.425	3.261	0.988	0.574	0.272
$\sigma^{1} / \sum_{\sigma} \sigma^{1}$	0.623	0.241	0.073	0.042	0.020
σ^{1}	8.425	3.261	0.988	0.574	0.272
σ^{2}	70.985	10.635	0.976	0.330	0.074
$\sigma^{2} / \sum_{\sigma} \sigma^{2}$	0.855	0.128	0.012	0.004	0.001
$\sigma^{1.5}$	24.455	5.889	0.982	0.435	0.142
σ^{3}	598.063	34.684	0.964	0.189	0.020
$\sigma^{3} / \sum_{\sigma} \sigma^{3}$	0.943	0.055	0.002	0.000	0.000

Figure 2 shows the two-dimensional plots of documents and terms for LSA-RAW with $\alpha = 0.5, 1.5$. The standard coordinates with $\alpha = 1$ was shown in Figure 1a. As α increases, the Euclidean distances between row points (column points) on the first dimension increase relative to the second dimension.



(a) $\alpha = 0.5$ (b) $\alpha = 1.5$

Fig. 2: A two-dimensional plot of documents and terms for LSA-RAW with (a) $\alpha = -0.5$ and (b) $\alpha = 1.5$.

3.3 Design

We compare the performances of LSA and CA for information retrieval, where two kinds of weightings are studied in LSA: the elements of the raw document-term matrix are weighted and the weighting exponent α is varied. We also explore the impact of these weightings in CA. We vary the number of dimension k from 1, 2, $\cdots$, 20, 22, $\cdots$, 50, 60, $\cdots$ to 100 and the value of α from -6, -5.5, $\cdots$, -2, -1.8, $\cdots$, 4, 4.5, $\cdots$ to 8; we explore all $40 \times 47 = 1,880$ combinations of parameter values.

In the study of weighting the elements of the raw document-term matrix, we perform the LSA and CA of

- raw matrix $\boldsymbol{F}$, denoted by RAW,
- L1 row-normalized matrix $\boldsymbol{F}^{L1}$ with $L(i,j) = f_{ij}$, $G(j) = 1$, and $N(i) = 1/\sum_{j=1}^{n} f_{ij}$, NROWL1,
- L2 row-normalized matrix $\boldsymbol{F}^{L2}$ with $L(i,j) = f_{ij}$, $G(j) = 1$, and $N(i) = 1/\sqrt{\sum_{j=1}^{n} f_{ij}^2}$, NROWL2, and
- TF-IDF matrix $\boldsymbol{F}^{\text{TF-IDF}}$ described in Section 2.1, TFIDF.

We refer to the combination of the CA and TF-IDF matrix as CA-TFIDF. Similarly, we obtain LSA-RAW, LSA-NROWL1, LSA-NROWL2, LSA-TFIDF, CA-RAW, CA-NROWL1, and CA-NROWL2. For performance comparison, RAW is used for term matchings without dimensionality reduction.

3.4 Datasets

LSA and CA are compared using three English datasets and one Dutch dataset. The three English datasets are the BBCSport (Greene and Cunningham, 2006), BBCNews (Greene and Cunningham, 2006), and 20 Newsgroups datasets (20-news-18846 bydata version) (Rennie, 2005). The Dutch dataset is the *Wilhelmus* dataset (Kestemont et al, 2017). The three English datasets have recently been used in information retrieval studies (Bounabi et al, 2019; Bianco et al, 2023). The *Wilhelmus* dataset is produced for studying authorship attribution of the song *Wilhelmus*, which is the national anthem of the Netherlands. The author of the song is unknown.

Some statistics of the four datasets used are presented in Table 3. The BBCNews dataset includes 2,225 documents that fall into one of five categories. The BBCSport dataset includes 731 documents that fall into one of five categories. The 20 Newsgroups dataset includes 18,846 documents that fall into one of 20 categories. This dataset is sorted into a training (60%) and a test (40%) set. We use a subset of this dataset to evaluate information retrieval. We randomly choose 600 documents from the training set of four categories (comp.graphics, rec.sport.hockey, sci.crypt, and talk.politics.guns) and 400 documents from the test set of these four categories. The *Wilhelmus* dataset includes 186 documents divided into six categories.

To pre-process the three English datasets, we change all characters to lower case, remove punctuation marks, numbers, and stop words, and apply lemmatization. Subsequently, terms with frequencies lower than 10 are ignored. In addition, we remove unwanted parts of the 20 Newsgroups dataset, such as the header (including fields like "From:" and "Reply-To:" followed by email address), because these are almost irrelevant for information retrieval. The Dutch *Wilhelmus* dataset is already pre-processed into tag-lemma pairs. Following Kestemont et al (2017) and Qi et al (2023), in *Wilhelmus* dataset, we use the 300 most frequent tag-lemma pairs.

Since the *Wilhelmus* and BBCSport datasets have a relatively low number of documents, we use leave-one-out cross-validation (LOOCV) for the

Table 3: Characteristics of datasets.

Categories	Data
business	510
entertainment	386
politics	417
sport	511
technology	401

(a) BBCNews dataset.

Categories	Data
athletics	101
cricket	124
football"	265
rugby"	147
tennis	100

(b) BBCSport dataset.

Categories	Training data	Test data
comp.graphics	141	100
rec.sport.hockey	164	99
sci.crypt	161	106
talk.politics.guns	134	95

(c) 20 Newsgroups dataset.

Categories	Data
datheen	35
marnix	46
heere	23
haecht	35
fruytiers	33
coornhert	14

(d) Wilhelmus dataset.

Wilhelmus dataset and five-fold cross-validation for the BBCSport dataset to evaluate LSA and CA (Gareth et al, 2021). The BBCNews dataset is randomly divided into training (80%) and validation (20%) sets.

In the information retrieval part of the study, each document in the validation set is used as a query, where the category of the document is known. The documents in the training set that fall in the same category as the query are the relevant documents for this query.

3.5 Evaluation

We compare the MAP of each of the four versions of LSA and CA to explore the performance of these methods in information retrieval under changes in the contributions of initial dimensions (Kolda and O'leary, 1998). The MAP is calculated as follows:

- The similarity is assessed between a query vector and each document vector of a document collection. We use three similarity metrics: Euclidean distance, dot similarity, and cosine similarity. As Euclidean distance is a key motivation for CA, we report results on Euclidean distance, and only report partial results for dot and cosine similarity in the main paper and the other results in the supplementary materials.
- For Euclidean distance, the documents are ranked in an increasing order based on their similarity with the query vector (for dot and cosine similarity, the ranking is in the decreasing order); therefore, the first document has the highest similarity.
- Precision-recall points are derived from the ordered list of documents. For a given query, Table 4 defines four types of documents in the ordered list based on whether a document is relevant and retrieved:
  C = the set of relevant documents from the ordered list, i.e., documents that fall in the same category as the query

$\mathbf{D}$ = the set of retrieved documents from the ordered list., i.e., when 10 documents are returned, the set of retrieved documents consists of the first 10 documents in the ordered list.

Table 4: Retrieved and relevant documents.

	Relevant	Non-Relevant
Retrieved	$\mathbf{C} \cap \mathbf{D}$	$\overline{\mathbf{C}} \cap \mathbf{D}$
Not Retrieved	$\mathbf{C} \cap \overline{\mathbf{D}}$	$\overline{\mathbf{C}} \cap \overline{\mathbf{D}}$

Let $|.|$ denote the number of documents in a set. Then, precision and recall are defined as

$$\text{precision} = \frac{|\mathbf{C} \cap \mathbf{D}|}{|\mathbf{D}|} \tag{11}$$

and

$$\text{recall} = \frac{|\mathbf{C} \cap \mathbf{D}|}{|\mathbf{C}|}. \tag{12}$$

Thus, precision is defined as the ratio of the number of relevant documents retrieved over the total number of retrieved documents, and recall is defined as the ratio of the number of relevant documents retrieved over the total number of relevant documents. For a given query, the set $\mathbf{C}$ is fixed. The set $\mathbf{D}$ is not fixed; if we return the first i documents, then $\mathbf{D}$ consists of the first i documents in the ordered list. Thus, for a given i, we can obtain a precision (see Equation (11)) and recall (see Equation (12)) pair. We run values of i from 1 to l (the number of documents in the ordered list), and obtain l precision-recall pairs.

- Then, 11 pseudo-precisions are calculated under 11 recalls (0, 0.1, $\cdots$, 1.0), where a pseudo-precision at recall x is the maximum precision from recall x to recall 1. For example, pseudo-precision at recall 0.2 is the maximum precision from recall 0.2 to recall 1.
- The average precision for the query is obtained by averaging the 11 pseudo-precisions.
- The MAP is the mean across all queries.

Greater MAP values indicate a better performance.

4 Results for Euclidean distance

4.1 Comparing LSA and CA for information retrieval

4.1.1 MAP as a function of the number of dimensions for the four versions of LSA with the standard weighting exponent $\alpha = 1$ and for CA

We first investigate the performance of LSA and CA in terms of MAP, in their standard use, i.e., without varying the weighting exponent α, i.e., $\alpha = 1$. Term matching without the preliminary use of LSA and CA, i.e., directly on

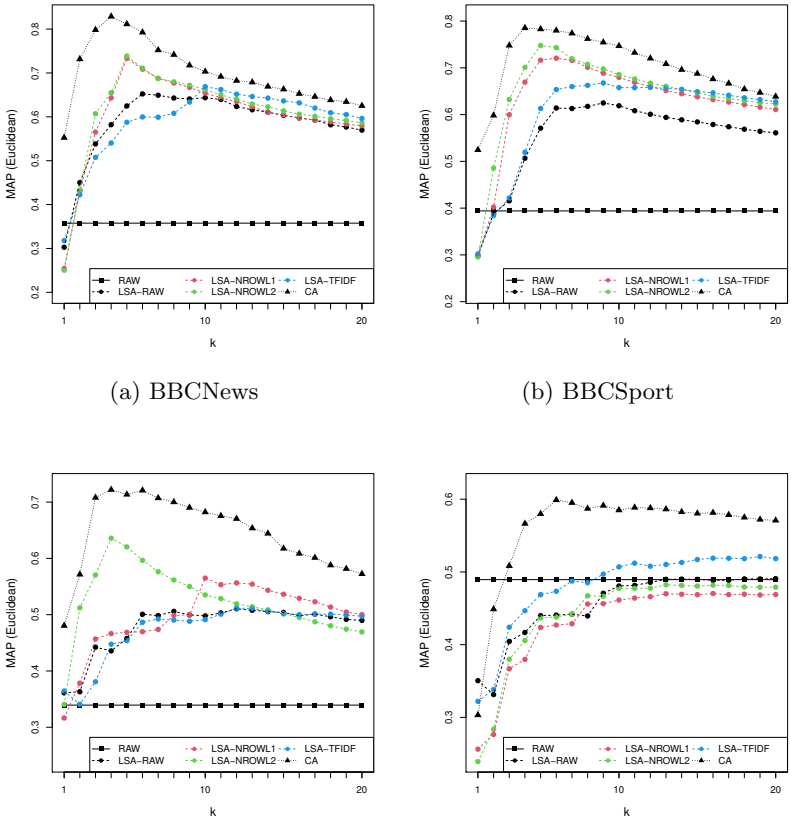
the document-term matrix, is denoted by RAW. We expect that, in line with Qi et al (2023), the performance of LSA and CA will be better than that of RAW, and the performance of CA will be better than that of the four versions of LSA.

Figure 3 shows MAP as a function of the number of dimensions k for different weighting schemes of LSA, and for CA. We display only the first 20 dimensions, as all lines usually decrease after dimension 20. Figures with dimensionality up to 100 can be found in the supplementary materials. For the four versions of LSA, and for CA, Table 5 presents the dimension number for which the optimal MAP is reached, as well as the MAP values, in each of the four datasets. We conclude the following from Figure 3 and Table 5:

- Both LSA and CA result in better MAP than RAW, which results in a straight line when the full dimensional matrix is used.
- For both LSA and CA, performance is a function of the number of dimensions k. Overall, MAP rises as a function of k to reach a peak, and then, it goes down. For CA, the peak is reached at $k = 4$. In CA, the information used to calculate MAP increases in the first four dimensions in comparison to the noise. In the components of $k \geq 5$, the noise dominates the useful information, which results in the MAP going down from this point.
- CA results in a considerably better MAP than the four versions of LSA: LSA-RAW, LSA-NROWL1, LSA-NROWL2, and LSA-TFIDF, which is in line with Qi et al (2023), who showed that the performance of CA is better than that of LSA for document-term matrices. This is because of the differential treatment of margins in LSA and CA. The margins provide irrelevant information for making queries. In CA, the margins are removed, and therefore, the relative amount of information in comparison to the noise, which we informally refer to as the information - noise ratio, is considerably larger in CA than in LSA. This explains the better MAP in CA.
- The peaks for the four versions of LSA are usually found at higher dimensionality k than the peaks for CA. This is because margins are noise for queries when we fix $\alpha = 1$; in LSA, this noise plays an important role in the first few dimensions. Hence, this earlier peak in CA is also explained by its better information - noise ratio.
- The four LSA methods are not equally effective. In all four datasets, the performance of LSA can be significantly improved using weighting schemes. The improvements over LSA-RAW are data dependent. On average, across the four datasets, LSA-NROWL2 is the best, but for the *Wilhelmus* dataset, LSA-NROWL1 and LSA-NROWL2 result in a somewhat worse MAP than that with LSA-RAW.

4.1.2 MAP as a function of the weighting exponent α for LSA compared with MAP for CA under varying numbers of dimensions

In Section 4.1.1, we found that CA outperforms the four versions of LSA in terms of MAP, where LSA had the usual weighting exponent $\alpha = 1$. In this



(a) BBCNews (b) BBCSport

(c) 20 Newsgroups (d) Wilhelmus

Fig. 3: MAP as a function of the number of dimensions k under standard coordinates.

section, we study whether the performance of LSA-RAW improves when we vary α.

Figure 4 shows MAP as a function of α for LSA-RAW with the number of dimensions $k = 4, 6, 9, 12,$ and 24. For comparison, we also report the MAP values for CA found in Section 4.1.1 under these dimensions. We choose these values of k because these dimensions are optimal for LSA-RAW and CA in Table 5. Table 6 shows the optimal α and corresponding MAP, which is a condensed version of Figure 4. We conclude the following from Figure 4 and Table 6:

- Although the performance of LSA-RAW improves by varying α, CA still outperforms LSA-RAW.

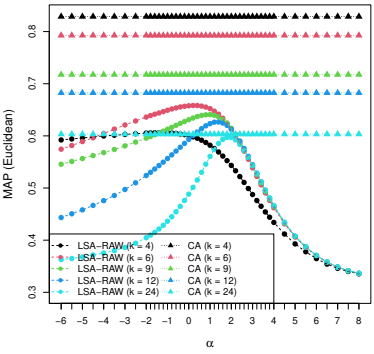
Table 5: MAP with the optimal number of dimensions k. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	k	MAP	k	MAP	k	MAP	k	MAP
RAW		0.358		0.394		0.339		0.489
LSA-RAW	6	0.652	9	0.625	12	0.510	24	0.492
LSA-NROWL1	5	0.733	6	0.721	10	0.565	16	0.470
LSA-NROWL2	5	0.738	5	0.748	4	0.636	13	0.482
LSA-TFIDF	10	0.669	9	0.668	12	0.512	19	0.521
CA	4	**0.829**	4	**0.785**	4	**0.722**	6	**0.599**

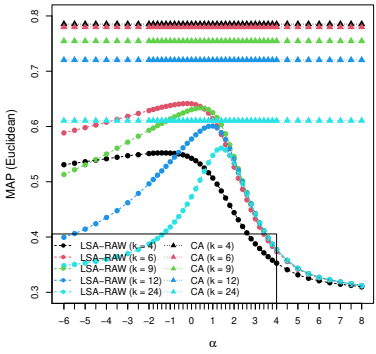
- For LSA-RAW, the overall MAP first increases and then decreases as a function of α. This means that varying α can potentially improve the performance of LSA-RAW.
- The increase in MAP is minor. Consider, for example, the BBCNews dataset. In Section 4.1.1, we found that the MAP was optimal with a value of 0.652 for $\alpha = 1$, when $k = 6$. Table 6 shows that for $\alpha = 0.2$, the MAP increases to 0.658. Apparently, for 6 dimensions, when $\alpha = 0.2$, the information - noise ratio is optimal in terms of MAP. For $\alpha = 0.2$, the distances on later dimensions (of the 6 dimensions) are increased and those on initial dimensions are reduced. This means that, with $\alpha = 0.2$, the impact of the initial dimensions affected most by the margins is reduced. This is consistent with the results of Bullinaria and Levy (2012), which indicates that reducing the initial dimensions improves performance.
- Moreover, the optimal α for LSA-RAW is data dependent and generally increases with k. This replicates results of Caron (2001). As the number of dimensions varies, the change in the optimal α is the result of the information - noise ratio for the specific number of dimensions studied. For example, for the BBCNews dataset, the optimal number of dimensions is 6; for larger numbers of dimensions, the optimal α increases. An increasing α indicates that distances at earlier dimensions are more important for information retrieval, and therefore, the role of the later dimensions is played down.

Table 6: MAP with the optimal weighting exponent α for LSA-RAW and MAP for CA under $k = 4, 6, 9, 12,$ and 24. Bold values are best.

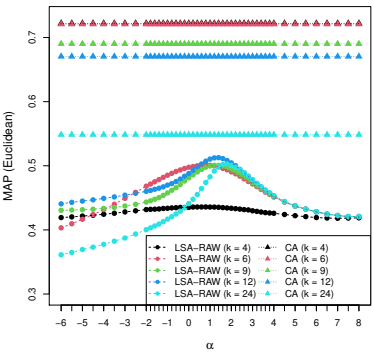
	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	α	MAP	α	MAP	α	MAP	α	MAP
LSA-RAW ($k = 4$)	-1.4	0.606	-1.4	0.552	0.8	0.436	0.2	0.424
LSA-RAW ($k = 6$)	0.2	0.658	-0.2	0.642	0.8	0.501	0.4	0.444
LSA-RAW ($k = 9$)	1	0.641	0.4	0.634	1.2	0.501	0.4	0.488
LSA-RAW ($k = 12$)	1.4	0.627	1	0.601	1.4	0.513	0.4	0.500
LSA-RAW ($k = 24$)	1.8	0.597	1.4	0.561	1.8	0.503	0.8	0.496
CA ($k = 4$)		**0.829**		**0.785**		**0.722**		0.566
CA ($k = 6$)		0.793		0.780		0.721		**0.599**
CA ($k = 9$)		0.717		0.755		0.690		0.591
CA ($k = 12$)		0.682		0.720		0.670		0.588
CA ($k = 24$)		0.603		0.611		0.548		0.563



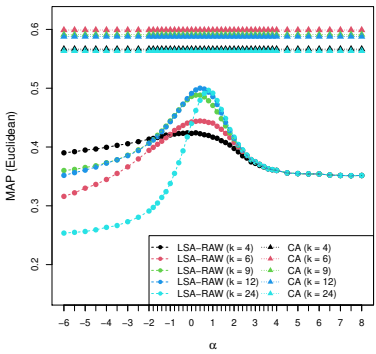
(a) BBCNews



(b) BBCSport



(c) 20 Newsgroups



(d) Wilhelmus

Fig. 4: MAP as a function of α for LSA-RAW and MAP for CA under varying k.

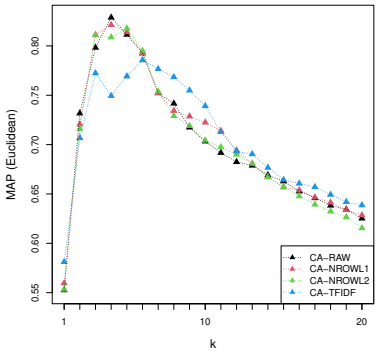
4.2 Adjusting CA using weighting

4.2.1 Weighting the elements of the raw document-term matrix for CA

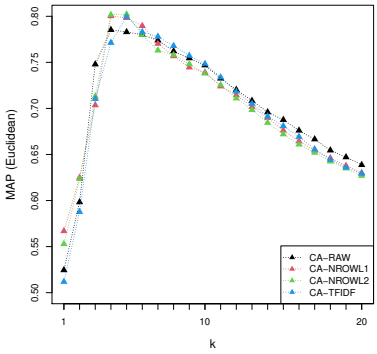
Weighting the elements of the raw document-term matrix is an effective way to improve the performance of LSA for information retrieval. Here, we explore whether this holds for CA. Similar to Figure 3, Figure 5 shows MAP as a function of k for different weighting schemes of CA. CA in Figure 3 is referred to as CA-RAW in Figure 5; for CA/CA-RAW, the results in these two figures are identical. For the four versions of CA, Table 7 shows the dimensionality for which the optimal MAP is reached, as well as the MAP value. We conclude the following from Figure 5 and Table 7:

- Overall, the weighting of the elements of the raw matrix sometimes improves the performance of CA, but these improvements over CA-RAW are small and data dependent.
- Comparing Table 5 with Table 7, the performance of CA-NROWL1 is better than that of LSA-NROWL1, the performance of CA-NROWL2 is better than that of LSA-NROWL2, and the performance of CA-TFIDF is better than that of LSA-TFIDF.

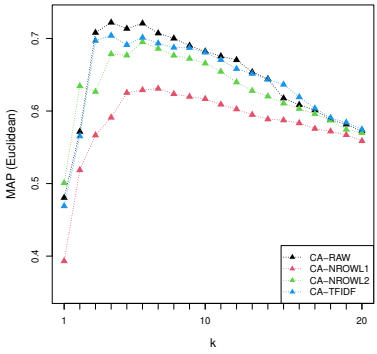
Relative to LSA, it is harder to improve the performance of CA in information retrieval by weighting the elements of the raw matrix because (1) the MAP of CA-RAW is already relatively high, and (2) CA-RAW has weighted the elements of the raw document-term matrix as it is an integral part of this technique (Equation (5)).



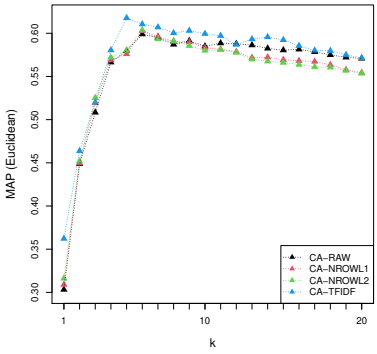
(a) $\alpha = 1$ (BBCNews)



(b) $\alpha = 1$ (BBCSport)



(c) $\alpha = 1$ (20 Newsgroups)



(d) $\alpha = 1$ (Wilhelmus)

Fig. 5: MAP as a function of the number of dimensions k for the four versions of CA under standard coordinates.

Table 7: MAP with the optimal number of dimensions k for the four versions of CA. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	k	MAP	k	MAP	k	MAP	k	MAP
CA-RAW	4	**0.829**	4	0.785	4	**0.722**	6	0.599
CA-NROWL1	4	0.821	4	0.800	7	0.631	6	0.603
CA-NROWL2	5	0.818	5	**0.802**	6	0.695	6	0.604
CA-TFIDF	6	0.786	5	0.800	4	0.704	5	**0.618**

4.2.2 MAP as a function of the weighting exponent α for CA

In this section, we introduce CA with weighting exponent α. Similar to Figure 4, Figure 6 shows MAP as a function of α in CA-RAW for the number of dimensions $k = 4, 6, 9, 12,$ and 24. Table 8 shows the optimal α and the corresponding MAP, which is a condensed version of Figure 6. We conclude the following from Figure 6 and Table 8:

- For CA, the overall MAP first increases and then decreases as a function of α. This means that varying α can potentially improve the performance of CA.
- The increase in MAP by adjusting α is data and dimension dependent.
- If we compare the maxima in Table 6 with those in Table 8 , there is hardly a noticeable increase.

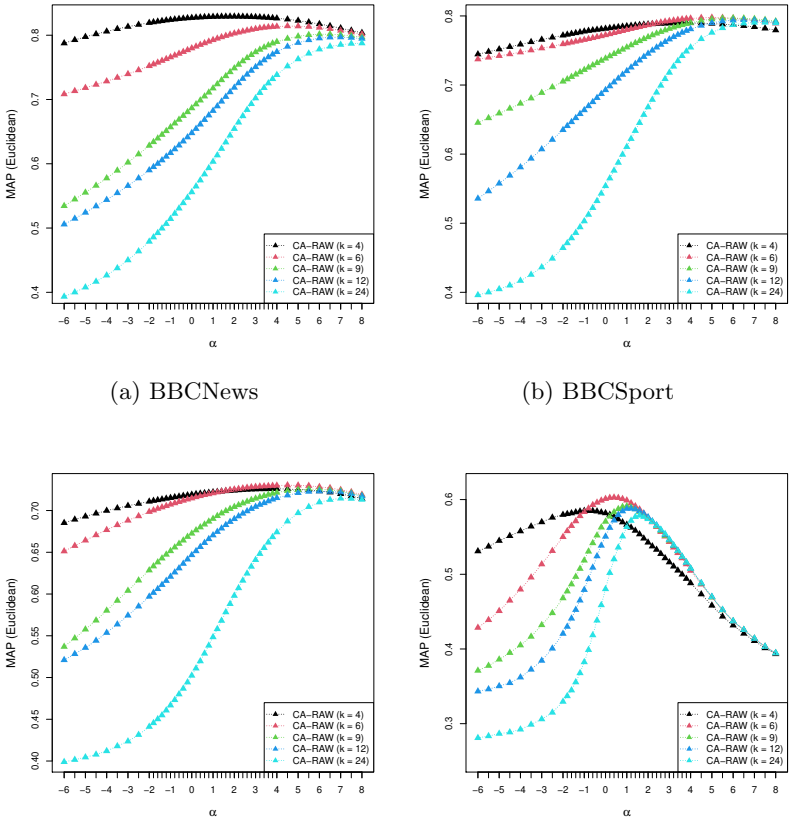
Now, we check the optimal α like Bullinaria and Levy (2012) did. Comparing Table 8 with part LSA-RAW of Table 6, the optimal α for CA-RAW is almost always larger than LSA-RAW and is almost always larger than 1. That is, CA-RAW needs a larger α than LSA-RAW to obtain its maximum MAP. Thus, compared to LSA, CA improves by placing more emphasis on its initial dimensions. The important difference between LSA and CA is that LSA involves margins, and CA does not. Therefore, we infer that margins in LSA considerably contribute to the initial dimensions; however, they are irrelevant ("noise") for information retrieval. On the other hand, CA effectively eliminates this irrelevant information.

We study MAP as a function of α under the optimal number of dimensions. The details including tables and figures are in the supplementary materials. Again, CA performs better than LSA. Adjusting α can potentially improve the performance of LSA and CA. Although the optimal α under the optimal number of dimensions is data dependent, the optimal α of CA is usually considerably larger than that of LSA.

5 Results for dot similarity and cosine similarity

In Section 4, we presented the results where Euclidean distance was used as a measure of similarity. Here, for comparison, we provide results for dot similarity and cosine similarity. Tables and figures for dot similarity and cosine similarity are presented in the supplementary materials.

The results for both dot similarity and cosine similarity lead to conclusions that match those for Euclidean distance. However, cosine similarity leads to a



(a) BBCNews (b) BBCSport

(c) 20 Newsgroups (d) Wilhelmus

Fig. 6: MAP as a function of α for CA-RAW under various values of k.

better performance in terms of MAP than Euclidean distance and dot similarity. We displayed the results for Euclidean distance in Section 4 because (1) it is more easily interpretable in the context of adjusting weighting exponent α: as α increases, Euclidean distances between row points (column points) on initial dimensions increase relative to the later dimensions; and (2) in the literature, the Euclidean distance is the preferred way to interpret CA (in fact, we have never seen an interpretation of CA in terms of cosine or dot similarity).

6 Conclusions and discussions

Both LSA and CA make use of SVD. The main difference between LSA and CA is the matrix that is decomposed by SVD. In LSA, the decomposed matrix is the weighted matrix $\boldsymbol{A}$. In CA the decomposed matrix is the matrix $\boldsymbol{S}$ of standardized residuals, where in the part $(\boldsymbol{P} - \boldsymbol{E})$ the marginal effects are eliminated (Qi et al, 2023), and whose rank is one less the rank of $\boldsymbol{A}$.

Table 8: MAP with the optimal α for CA-RAW under $k = 4, 6, 9, 12,$ and 24. Bold values are best.

	BBCNews		BBCSport		20 Newsgroups		Wilhelmus	
	α	MAP	α	MAP	α	MAP	α	MAP
CA-RAW ($k = 4$)	2	**0.829**	3.6	0.790	4	0.726	-1	0.585
CA-RAW ($k = 6$)	4.5	0.814	5	**0.798**	4.5	**0.730**	0.4	**0.603**
CA-RAW ($k = 9$)	6.5	0.802	6	0.797	5.5	0.726	1	0.591
CA-RAW ($k = 12$)	7	0.797	6.5	0.794	6	0.723	1.2	0.588
CA-RAW ($k = 24$)	8	0.788	7.5	0.791	7	0.715	1.6	0.579

That is why the CA solution only displays the dependence between documents and terms. In LSA, on the other hand, the decomposed matrix also includes marginal effects, which are usually not relevant for information retrieval.

CA is related to the statistical independence model (Greenacre, 1984). The elements of $\boldsymbol{S}$ display the departure from marginal products, i.e., the departure form the statistical independence model. The sum of squared elements of $\boldsymbol{S}$ equals the Pearson chi-square statistic divided by the sum of elements of $\boldsymbol{F}$. CA decomposes the departure from statistical independence into a number of dimensions using SVD. LSA, on the other hand, has no connection with the statistical independence model.

In this paper, we compared four versions of LSA: LSA-RAW, LSA-NROWL1, LSA-NROWL2, and LSA-TFIDF with CA and found that CA always performs better than LSA in terms of MAP. Then, we compared LSA-RAW as a function of weighting exponent α with CA under a range of the numbers of dimensions. Even though LSA is improved by choosing an appropriate value for α, CA always performed better than LSA.

Next, we applied different weighting elements of the raw document-term matrix to CA. We found that weighting elements of the raw matrix sometimes improves the performance of CA, but improvements over CA-RAW are small and data dependent. The performance of CA-NROWL1 is better than that of LSA-NROWL1, the performance of CA-NROWL2 is better than that of LSA-NROWL2, and the performance of CA-TFIDF is better than that of LSA-TFIDF. Then, we adjusted the weighting exponents α in CA. For CA, as a function of α, MAP first increases and then decreases. Adjusting the weighting exponent α can potentially improve the performance of CA. However, the increased performance obtained by adjusting α is data and dimension dependent.

Using the standard coordinates of $\alpha = 1$, for LSA, the Euclidean distances between the rows of coordinates approximate the Euclidean distances between the rows of the decomposed matrix. For CA, the Euclidean distances between the rows of coordinates approximate the χ^2-distances between the rows of the decomposed matrix. $\alpha < 1$ gives less emphasis to the initial dimensions relative to the standard coordinates. Conversely, $\alpha > 1$ gives more emphasis to the initial dimensions relative to the standard coordinates. The optimal α for CA is almost always larger than that for LSA and is almost always larger than 1.

Bullinaria and Levy (2012) argued that the initial dimensions in LSA tend not to contribute the most useful information about semantics and tend to be contaminated by "noise". The above mentioned results indicate that CA places more emphasis on the initial dimensions than LSA. The major difference between LSA and CA is that LSA involves margins but CA does not (Qi et al, 2023). Thus, we infer that margins considerably contribute to the initial dimensions in LSA. These margins are irrelevant for information retrieval. The CA effectively eliminates this irrelevant information.

In this paper, we focused on the performances of CA and LSA using Euclidean distances. We also performed identical experiments for dot similarity and cosine similarity. Both have nearly identical results with the Euclidean distance. Cosine similarity performs better than the Euclidean distance and dot similarity. We focus on Euclidean distance in the paper because (1) it is more easily interpretable in the context of adjusting α: as α increases, the Euclidean distances between row points (column points) on the initial dimensions increase relative to the later dimensions; (2) for CA, dot similarity and cosine similarity have never been used before, and therefore, by focusing on Euclidean distances, the results fit better into the existing literature.

Based on theoretical considerations and experimental results, we have the following three suggestions for practical guidance:

1. Use CA instead of LSA under the four kinds of feature extraction: RAW, NROWL1, NROWL2, and TF-IDF; use CA for visualizing data.
2. If information retrieval is the key issue, use cosine similarity instead of Euclidean distance and dot similarity for calculating MAP.
3. If optimal performance in terms of MAP is not of key importance, there is no need to weight the elements of raw document-term matrix for CA and optimize the performance over α for CA to saving time. Otherwise, these two weightings may be considered potential approaches for improving the performance of CA.

Our finding that CA performs better than LSA for information retrieval is very important for creating next generation intelligent information systems. Among many other tasks, LSA has been widely used for information retrieval. We expect that the performance of these tasks can be improved by replacing LSA with CA.

Concluding, CA and LSA are both tools for information retrieval but the performance of CA is better. In our paper we tried to further improve CA by weighting the input matrix and by weighting dimensions. This did not lead to large or consistent improvements of the performance of CA.

Further studies on the combination of LSA and CA will also be interesting. For example, creating an ensemble voting system using the coordinates from LSA and CA in the process of returning documents of a query. This paper, however, focuses on the comparison of LSA and CA for information retrieval and other explorations are left for future studies.

Acknowledgments

Author Qianqian Qi is supported by the China Scholarship Council.

Declarations

Ethical Approval

Not applicable

Competing interests

Author Qianqian Qi is supported by the China Scholarship Council (CSC202007720017). Author David J. Hessen and Author Peter G. M. van der Heijden have no competing interests to declare that are relevant to the content of this article.

Authors' contributions

Author Qianqian Qi posed the problem and set up the experiments. Author Qianqian Qi, Author David J. Hessen, and Author Peter G. M. van der Heijden discussed and edited the text.

Funding

Author Qianqian Qi is supported by the China Scholarship Council (CSC202007720017).

Availability of data and materials

In this article, the BBCNews dataset, BBCSport dataset, 20 Newsgroups dataset, and *Wilhelmus* dataset are used.

The BBCNews dataset that supports the findings of this article is available at http://mlg.ucd.ie/datasets/bbc.html.

The BBCSport dataset that supports the findings of this article is available at http://mlg.ucd.ie/datasets/bbc.html

The 20 Newsgroups dataset that supports the findings of this article is available at http://qwone.com/~jason/20Newsgroups/

The *Wilhelmus* dataset that supports the findings of this article is available in GitHub at https://github.com/mikekestemont/anthem

References

Aggarwal CC (2018) Machine learning for text. Springer, https://doi.org/https://doi.org/10.1007/978-3-319-73531-3

Al-Qahtani M, Amira A, Ramzan N (2015) An efficient information retrieval technique for e-health systems. In: 2015 International Conference on Systems, Signals and Image Processing (IWSSIP), 257–260, https://doi.org/10.1109/IWSSIP.2015.7314225

Altszyler E, Sigman M, Ribeiro S, et al (2016) Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database. Preprint at https://arxiv.org/abs/1610.01520

Arenas-Márquez FJ, Martinez-Torres R, Toral S (2021) Convolutional neural encoding of online reviews for the identification of travel group type topics on tripadvisor. Information Processing & Management 58(5):102,645. https://doi.org/https://doi.org/10.1016/j.ipm.2021.102645

Azmi AM, Al-Jouie MF, Hussain M (2019) AAEE–Automated evaluation of students' essays in Arabic language. Information Processing & Management 56(5):1736–1752. https://doi.org/https://doi.org/10.1016/j.ipm.2019.05.008

Bacciu A, Morgia ML, Mei A, et al (2019) Bot and Gender Detection of Twitter Accounts Using Distortion and LSA. In: CLEF

Baroni M, Bernardini S, Ferraresi A, et al (2009) The wacky wide web: a collection of very large linguistically processed web-crawled corpora. Language Resources and Evaluation 43(3):209–226. https://doi.org/https://doi.org/10.1007/s10579-009-9081-4

Beh EJ, Lombardo R (2021) An introduction to correspondence analysis. John Wiley & Sons

Berry MW, Dumais ST, O'Brien GW (1995) Using linear algebra for intelligent information retrieval. SIAM Review 37(4):573–595. https://doi.org/https://doi.org/10.1137/1037127

Bianco GD, Duarte D, Gonçalves MA (2023) Reducing the user labeling effort in effective high recall tasks by fine-tuning active learning. Journal of Intelligent Information Systems https://doi.org/10.1007/s10844-022-00772-y

Bounabi M, Moutaouakil KE, Satori K (2019) A comparison of text classification methods using different stemming techniques. International Journal of Computer Applications in Technology 60(4):298–306. https://doi.org/10.1504/IJCAT.2019.101171

Bullinaria JA, Levy JP (2012) Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. Behavior Research Methods 44(3):890–907. https://doi.org/https://doi.org/10.3758/s13428-011-0183-8

Caron J (2001) Experiments with LSA scoring: Optimal rank and basis. In: Proceedings of the SIAM Computational Information Retrieval Workshop, 157–169

Chang CY, Lee SJ, Wu CH, et al (2021) Using word semantic concepts for plagiarism detection in text documents. Information Retrieval Journal

24:298–321. https://doi.org/https://doi.org/10.1007/s10791-021-09394-4

Deerwester S, Dumais ST, Furnas GW, et al (1990) Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6):391–407. https://doi.org/https://doi.org/10.1002/(SICI)1097-4571(199009)41:6⟨391::AID-ASI1⟩3.0.CO;2-9

Drozd A, Gladkova A, Matsuoka S (2016) Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 3519–3530, URL https://aclanthology.org/C16-1332

Duan L, Gao T, Ni W, et al (2021) A hybrid intelligent service recommendation by latent semantics and explicit ratings. International Journal of Intelligent Systems 36(12):7867–7894. https://doi.org/https://doi.org/10.1002/int.22612

Dumais ST (1991) Improving the retrieval of information from external sources. Behavior Research Methods, Instruments, & Computers 23(2):229–236. https://doi.org/https://doi.org/10.3758/BF03203370

Dumais ST, Furnas GW, Landauer TK, et al (1988) Using latent semantic analysis to improve access to textual information. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 281–285, https://doi.org/https://doi.org/10.1145/57167.57214

Gabriel KR (1971) The biplot graphic display of matrices with application to principal component analysis. Biometrika 58(3):453–467. https://doi.org/https://doi.org/10.2307/2334381

Gareth J, Daniela W, Trevor H, et al (2021) An introduction to statistical learning: with applications in R. Springer

Greenacre MJ (1984) Theory and applications of correspondence analysis. Academic Press

Greenacre MJ (2017) Correspondence analysis in practice. CRC Press

Greene D, Cunningham P (2006) Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proceedings of the 23rd International Conference on Machine Learning, 377–384, https://doi.org/https://doi.org/10.1145/1143844.1143892

Guo J, Cai Y, Fan Y, et al (2022) Semantic models for the first-stage retrieval: A comprehensive review. ACM Transactions on Information Systems (TOIS) 40(4):1–42. https://doi.org/https://doi.org/10.1145/3486250

Gupta H, Patel M (2021) Method Of Text Summarization Using Lsa And Sentence Based Topic Modelling With Bert. In: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 511–517, https://doi.org/10.1109/ICAIS50930.2021.9395976

Hassani A, Iranmanesh A, Mansouri N (2021) Text mining using nonnegative matrix factorization and latent semantic analysis. Neural Computing and Applications 33:13,745–13,766. https://doi.org/https://doi.org/10.1007/s00521-021-06014-6

Horasan F (2022) Latent Semantic Indexing-Based Hybrid Collaborative Filtering for Recommender Systems. Arabian Journal for Science and Engineering 47:10,639–10,653. https://doi.org/https://doi.org/10.1007/s13369-022-06704-w

Horasan F, Erbay H, Varçın F, et al (2019) Alternate Low-Rank Matrix Approximation in Latent Semantic Analysis. Scientific Programming 2019:1–12. https://doi.org/https://doi.org/10.1155/2019/1095643

Hou R, Huang CR (2020) Classification of regional and genre varieties of chinese: A correspondence analysis approach based on comparable balanced corpora. Natural Language Engineering 26(6):613–640. https://doi.org/10.1017/S1351324920000121

Hu X, Cai Z, Franceschetti D, et al (2003) LSA: First dimension and dimensional weighting. In: Proceedings of the Annual Meeting of the Cognitive Science Society

Kestemont M, Stronks E, De Bruin M, et al (2017) Retrieved July 17, 2021, from https://github.com/mikekestemont/anthem

Kolda TG, O'leary DP (1998) A semidiscrete matrix decomposition for latent semantic indexing information retrieval. ACM Transactions on Information Systems (TOIS) 16(4):322–346. https://doi.org/https://doi.org/10.1145/291128.291131

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. https://doi.org/10.1162/tacl_a_00134

Liu T, Ungar L, Sedoc J (2019) Unsupervised post-processing of word vectors via conceptor negation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 6778–6785, https://doi.org/10.1609/aaai.v33i01.33016778

Morin A (2004) Intensive use of correspondence analysis for information retrieval. In: 26th International Conference on Information Technology Interfaces, 2004, 255–258

Mu J, Viswanath P (2018) All-but-the-top: Simple and effective post-processing for word representations. 6th International Conference on Learning Representations, ICLR 2018

Österlund A, Ödling D, Sahlgren M (2015) Factorization of latent variables in distributional semantic models. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 227–231, https://doi.org/10.18653/v1/D15-1024

Paralı U, Zontul M, Ertuğrul DÇ (2019) Information retrieval using the reduced row echelon form of a term-document matrix. Journal of Internet Technology https://doi.org/10.3966/160792642019072004004

Patil A (2022) Word Significance Analysis in Documents for Information Retrieval by LSA and TF-IDF using Kubeflow. In: Expert Clouds and Applications. Springer Singapore, Singapore, 335–348, https://doi.org/https://doi.org/10.1007/978-981-16-2126-0_29

Phillips T, Saleh A, Glazewski KD, et al (2021) Comparing Natural Language Processing Methods for Text Classification of Small Educational Data. In: Companion Proceedings 11th International Conference on Learning Analytics & Knowledge

Qi Q, Hessen DJ, Deoskar T, et al (2023) A comparison of latent semantic analysis and correspondence analysis of document-term matrices. Natural Language Engineering 1–31. https://doi.org/10.1017/S1351324923000244

Rennie J (2005) 20 newsgroups data set. Retrieved April 21, 2022, from http://qwone.com/~jason/20Newsgroups/

Séguéla J, Saporta G (2011) A comparison between latent semantic analysis and correspondence analysis. In: CARME 2011 International Conference on Correspondence Analysis and Related Methods

Suleman RM, Korkontzelos I (2021) Extending latent semantic analysis to manage its syntactic blindness. Expert Systems with Applications 165:114,130. https://doi.org/https://doi.org/10.1016/j.eswa.2020.114130

Van Dam A, Dekker M, Morales-Castilla I, et al (2021) Correspondence analysis, spectral clustering and graph embedding: applications to ecology and economic complexity. Scientific Reports 11(1):1–14. https://doi.org/https://doi.org/10.1038/s41598-021-87971-9

Yin Z, Shen Y (2018) On the dimensionality of word embedding. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, NIPS'18, 895–906

Zhang W, Yoshida T, Tang X (2011) A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Systems with

Applications 38(3):2758–2765. https://doi.org/https://doi.org/10.1016/j.eswa.2010.08.066