CHENNAI
MATHEMATICAL
INSTITUTE

Matrices, Vector spaces and Information Retrieval

Project report

Sneha KK(MDS202240)
Soham(MDS202241)
Souradeep(MDS202242)
Subhashree(MDS202243)

Msc Data Science semester project under the supervision of
Dr Priyavrat Deshpande,
Assistant professor,
Chennai Mathematical Institute

April 2023

Member	Contributions
Subhashree	Studied and implemented the vector space model, query matching,term-document comparison and QR factorization.
Soham	Studied and implemented QR Factorization with an Example of term-document comparison,its advantage and geometry, and low rank approximation.
Sneha KK	Studied and implemented SVD for term-document comparison with an example and query matching.
Souradeep	Studied and implemented the Term-Term comparison using an example,drew the conclusions and limitations from the paper.

Table 1: Contributions of team members

1.Introduction

The digitization of data has provided new challenges in storing and indexing information. As the Internet grows, massive amounts of data need to be efficiently indexed. One method for accomplishing this uses the power of vector spaces . The purpose of this paper is to implement information retrieval with the vector space model. Data are modeled as a matrix, and a user's query of the database is represented as a vector. Relevant documents in the database are then identified via simple vector operations.Then we will explore more advanced applications of linear algebra like QR factorization and Singular Value Decomposition (SVD) which can be used to manage and index large text collections and finally follow with a practical example.

1.1 Challenges in Indexing Large Collection of Information

A. Capacity or scale
1. Periodicals worldwide
i. 160,000 already in print
ii. 12,000 being added per year

2. Books in the U.S. alone:
i. 1.4 million already in print
ii. 60,000 being added per year

B. Indexing Inconsistencies
1.Indexing starts by selecting a list of "terms" to be used to classify documents for a given database.
2 Extraction of concepts and keywords from documents for indexing is intrinsically a fragile process.
3 An average of 20% disparity in the terms chosen as appropriate to describe a given document by any 2 different professional indexers.
4 Selecting such a list by indexer depends on the experience and opinions of the indexer such as:

i. age
ii. cultural background
iii. education
iv. political orientation
v. language - languages have quite a bit of ambiguities due to polysemy (words having multiple meanings) and synonymy (multiple words having the same meaning).

2.Vector space model

2.1 Vector space representation of Information

In the vector space IR model, a vector is used to represent each item or document in a collection. Each component of the vector reflects a particular concept, key word, or term associated with the given document.

A database containing a total of d documents described by t terms is represented as a t × d term-by-document matrix A. The d vectors representing the d documents form the columns of the matrix. Thus, the matrix element a_{i_j} is the weighted frequency at which term i occurs in document j. In the parlance of the vector space model, the columns of A are the document vectors, and the rows of A are the term vectors.

In a vector space IR scheme, a user queries the database to find relevant documents, somehow using the vector space representation of those documents. Query matching is finding the documents most similar to the query in use and weighting of terms. In the vector space model, the documents selected are those geometrically closest to the query according to some measure.One common measure of similarity is the cosine of the angle between the query and document vectors. If the term-by-document matrix A has columns a_j , j = 1,...,d , those d cosines are computed according to the formula:

$$cos\theta_j = \frac{a_j^t q}{||a_j||_2||q_i||_2}$$

here a_j represents the jth column of term-by-document matrix A, $j \in 1, 2, .., d$ and q represents the t-vector representation entered by the user.

2.2 An example

The t=6 terms:

T1: Cigarette(s)
T2:Smok(e,ing)
T3:lung(s)
T4:Cancer
T5: Vap(e,ing)
T6:Study

The d=5 document titles:

D1: Does cigarette smoking/vaping cause lung cancer?: a case study
D2: How to smoke lamb: a guide for new chefs
D3: The emergence of vaping : a new trend
D4:How cigarette smoking affects lungs
D5:Vaping or smoking: heated debate of 21st century

A collection of five titles is described by six terms. This leads to a 6×5 term-by-document matrix.
The 6 × 5 term-by-document matrix before normalization, where the element a_{ij} is the number of times term i appears in document title j:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\text{Normalized matrix :} \begin{pmatrix} 0.4082 & 0 & 0 & 0.5774 & 0 \\ 0.4082 & 1 & 0 & 0.5774 & 0.7071 \\ 0.4082 & 0 & 0 & 0.5774 & 0 \\ 0.4082 & 0 & 0 & 0 & 0 \\ 0.4082 & 0 & 1 & 0 & 0.7071 \\ 0.4082 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2.3 Query matching

Suppose we want to access documents related to vaping. So, query vector would be q = $(000010)^t$. For this query, the cosines of the angles between the query and five documents are given as follows:

$\cos\theta_1$	0.4082
$\cos\theta_2$	0
$\cos\theta_3$	1
$\cos\theta_4$	0
$\cos\theta_5$	0.7071

The jth document is returned as relevant to the query only if cos θ_j is greater than a certain threshold value.In this simple example, we will use 0.5. Thus the third and the last book is returned. The interesting consequence is that the first book, which is in fact a more comprehensive reference book about vaping, is not returned as relevant! Such failures can be alleviated by replacing A by a low-rank approximation to remove noise and uncertainties from the database representation.

3.QR Factorization

The QR factorization can be used to identify and remove redundant information in the matrix representation of the database.In linear algebra terms, we identify the rank of the term-by-document matrix and then try to lower the rank of the matrix further by removing components that can be attributed to the natural uncertainties present in any large database. The rank-reduction steps allow us to set portions of the matrix to zero and thus to ignore them in subsequent computations.

For a rank r_A matrix, the r_A basis vectors of its column space serve in place of its d column vectors to represent its column space. One set of basis vectors is found by computing the QR factorization of the term-by-document matrix

$$\mathrm{A} = \mathrm{QR},$$

where R is a t × d upper triangular matrix and Q is a t × t orthogonal matrix. The relation A = QR shows that the columns of A are all linear combinations of the columns of Q. Thus, a subset of r_A of the columns of Q forms a basis for the column space of A.If A = QR, the factors are:

$$\mathrm{Q}=\left(\begin{array}{cccc|cc} -0.4082 & 0.1825 & -0.2236 & 0.5 & 0.7071 & 0 \\ -0.4082 & -0.9128 & 0 & 0 & 0 & 0 \\ -0.4082 & 0.1825 & -0.2236 & 0.5 & -0.7071 & 0 \\ -0.4082 & 0.1825 & -0.2236 & 0.5 & 0 & -0.7071 \\ -0.4082 & -0.1825 & 0.8944 & 0 & 0 & 0 \\ -0.4082 & 0.1825 & -0.2236 & -0.5 & 0 & 0.7071 \end{array}\right)$$

and

$$R=\left(\begin{array}{ccccc} -1 & -4082 & -4082 & -7071 & -0.5774 \\ 0 & -0.9128 & 0.1825 & -0.3162 & -0.5164 \\ 0 & 0 & 0.8944 & -0.2581 & 0.6324 \\ 0 & 0 & 0 & 0.5774 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

We now rewrite the factorization A = QR as

$$A=\begin{pmatrix} Q_A & Q_A^{\perp} \end{pmatrix} \begin{pmatrix} R_A \\ 0 \end{pmatrix}$$
$$=Q_A R_A + Q_A^{\perp}.0 = Q_A R_A$$

The above partitioning clearly reveals that the columns of $Q_A^{\perp}$ do not contribute to the value of A and that the ranks of A, R and RA are equal. The semantic content of a database is fully described by any basis for the column space of the associated term-by-document matrix, and query matching proceeds with the factors QR in place of matrix A.

The j cosines, j $\in 1, 2, .., d$, are now calculated as:
$cos\theta_j = \frac{a_j^t q}{||a_j||_2||q||_2} = \frac{(Q_A r_j)^t q}{||Q_A r_j||_2||q||_2} = \frac{(Q_A q)^t (r_j)^t}{||r_j||_2||q||_2}$ for j = 1, . . . , d.

The cosines obtained by this formula are same as those computed earlier: 0.4082,0,1,0,0.7071.So, we observe that there is no loss of information in using its factored form.

3.1 More geometry

The QR factorization allows us a new geometrical interpretation of the vector space. We know,

$$I = QQ^T = (Q_A\ Q_A^{\perp})(Q_A\ Q_A^{\perp})^T = Q_A Q_A^T + Q_A^{\perp}(Q_A^{\perp})^T$$

Using the properties of orthogonal matrices,

$q = Iq = QQ^T q = Q_A Q_A^T q + Q_A^{\perp}(Q_A^{\perp})^T q = q_A + q_A^{\perp}$..................(i)
where q_A and $q_A^{\perp}$ are the projections of q onto the spaces of Q_A and $Q_A^{\perp}$ respectively.

Substituting this result produces
$cos\theta_j = \frac{a_j^T (q_A + q_A^{\perp})}{||a_j||_2||q||_2}$

from (i) $q_A^{\perp} = Q_A^{\perp}(Q_A^{\perp})^T q$, $cos\theta_j = \frac{a_j^T q_A + a_j^T Q_A^{\perp}(Q_A^{\perp})^T q}{||a_j||_2||q||_2}$
$=> cos\theta_j = \frac{a_j^T q_A + 0*(Q_A^{\perp})^T q}{||a_j||_2||q||_2} = \frac{a_j^T q_A}{||a_j||_2||q||_2}$

Taking the above equation a little further,

$$cos\theta'_j = \frac{a_j^T q_A}{||a_j||_2||q_A||_2}$$

Now we are using the projection of the query to compare with each document. Examining a comparison between these two different methods of searching leads to

$$cos\theta_j = cos\theta'_j \frac{||q_A||}{||q||}$$

We have $q_A^\perp = q - q_A$.Then from the geometry of the projection we can use the Pythagorean Theorem to say

$$cos\theta'_j \frac{||q_A||}{\sqrt{||q_A||^2+||q_A^\perp||^2}}$$

This new formula may return more documents.These documents might be relevant, which is good, or they may be irrelevant, which is bad. Either way this method has the potential to better the search.But in our example it returns the same cosines as computed earlier.

4.Low Rank approximation using QR

Indexing the database can lead to uncertainty in the term-by-document matrix. Hence, a term-by-document matrix A might be better represented by a matrix sum A+E, where the uncertainty matrix E may have any number of values reflecting missing or incomplete information about documents or even different opinions on the relevancy of documents to certain subjects. If we assume that our matrix A is only one representative of a whole family of relatively close matrices representing the database, it is reasonable to exactly determine its rank. But by adding a small change E would result in a matrix A+E of lesser rank. Hence, lowering the rank may help to remove extraneous information or noise from the matrix representation of the database. Our aim is to find a reasonable low-rank approximation to the matrix A. As rank of A is equal to R, we focus on the upper triangular matrix R as:

$$\text{R= Q=} \left(\begin{array}{cccc|c} -1.0001 & 0 & -0.5774 & -7070 & -0.4082 \\ 0 & -1.0000 & 0 & -4082 & -0.7071 \\ 0 & 0 & 0.8165 & 0 & 0.5774 \\ 0 & 0 & 0 & -0.5774 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

We now create a new upper triangular matrix R' by setting the small matrix R_{22} equal to the zero matrix. The new matrix R' has rank 3, as does the matrix

A + E = QR'. The uncertainty matrix E is then given by the difference

E = (A + E) - A =Q $\begin{pmatrix} 0 & 0 \\ 0 & -R_22 \end{pmatrix}$

now, $||E||_F = ||R_{22}||_F, ||A||_F = ||R||_F$

$\frac{||E||_F}{||A||_F} = \frac{||R_{22}||_F}{||R||_F} = \frac{0.5774}{2.2361} = 0.2582$

Hence, it is making a 26 percent relative change in the value of R makes the same-sized change in A,and that change reduces the ranks of both matrices by 1.Thus, we may deem it acceptable to use the rank-3 approximation A + E in place of the original term-by-document matrix A for query matching.

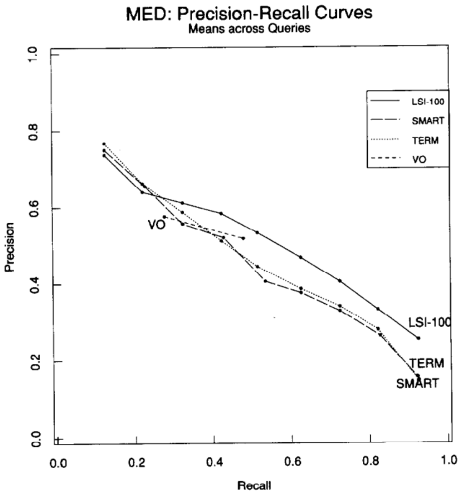
5. Singular value decomposition

5.1 The need for SVD

The vector space model using QR factorization gives us an approximate to the basis of column space only. However, to compare the rows (terms) we need a method that gives us an idea about the row space. This is precisely why we resort to SVD: it also gives us a reduced rank basis for the row space. Although SVD is computationally more expensive than QR, it has a lot of advantages. It also gives us the best rank k approximation to our matrix A for a given k and term by term comparison helps us deal with polysemy and synonymy (hence increasing precision and recall).

5.2 MED data set

The paper [1] on Latent Semantic Analysis compares our Latent Semantic Indexing with SVD approach with other standard approaces (SMART , TERM , and Voorhes) on the MED data set , a commonly studied collection of medical abstracts. It consists of 1033 documents and 30 queries. Precision was measured by varying recall and plotted:



According to the figure, our LSI approach gives us much better precision upon increasing recall than other methods. Having emphasized the importance of SVD, let's get into the mathematical construct.

5.3 SVD

The singular value decomposition of a $t \times d$ matrix A is written as:

$$A = U\Sigma V^T \text{ where}$$
$$U : t \times t \text{ orthogonal matrix of left singular vectors}$$
$$V : d \times d \text{ orthogonal matrix of right singular vectors}$$
$$\Sigma : t \times d \text{ diagonal matrix of singular values } \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{min(t,d)}$$

The rank r of A is the number of nonzero singular values of A , and the first r left singular vectors form a basis for the column space of A while the first r right singular vectors form a basis for the row space of A.For any k, the rank k approximation A_k can be written as $A_k = U_k \Sigma_k {V_k}^T$ where U_k is the $t \times k$ matrix of first k columns of U, V_k is the $k \times d$ matrix of first k columns of V and Σ_k is the $k \times k$ diagonal matrix of k largest singular values.

The Frobenius norm of A can be written as:

$$\parallel A \parallel_F = \parallel U\Sigma V^T \parallel_F = \parallel \Sigma \parallel_F$$

by the orthogonal invariance of the Frobenius norm. This is further equal to $\sqrt{\sum\limits_{j=1}^{r} \sigma_j^2}$. Similarly $\parallel A_k \parallel = \sqrt{\sum\limits_{j=1}^{k} \sigma_j^2}$.

5.4 Eckart's theorem

The distance between A and its rank k approximation is minimized by $A_k = U_k \Sigma_k V_k^T$ (ie.) SVD gives the best rank k approximation.

$$\begin{gathered}
\text{Norm of distance} = \parallel A - A_k \parallel_F \\
= \min_{rank(X) \leq k} \parallel A - X \parallel_F \\
= \parallel U \Sigma V^T - U_k \Sigma_k V_k^T \parallel_F = \parallel U' \Sigma' V_T{}' \parallel_F \\
= \parallel \Sigma' \parallel_F = \sqrt{\sigma_{k+1}^2 + \cdots \sigma_r^2}\ .
\end{gathered}$$

This gives us the minimum distance as the last few singular vectors give the direction of minimum change and the last few singular values give the magnitude. Our term document matrix gave the following SVD:

$$\text{A} = \begin{pmatrix} 0.4082 & 0 & 0 & 0.5774 & 0 \\ 0.4082 & 1 & 0 & 0.5774 & 0.7071 \\ 0.4082 & 0 & 0 & 0.5774 & 0 \\ 0.4082 & 0 & 0 & 0 & 0 \\ 0.4082 & 0 & 1 & 0 & 0.7071 \\ 0.4082 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0.27 & -0.26 & 0.53 & -0.28 & 0.71 & 0 \\ 0.75 & -0.4 & -0.52 & 0.08 & 0 & 0 \\ 0.26 & -0.26 & 0.53 & -0.29 & -0.71 & 0 \\ 0.12 & -0.01 & 0.28 & 0.64 & 0 & -0.71 \\ 0.52 & 0.84 & 0.08 & 0.12 & 0 & 0 \\ 0.12 & -0.01 & 0.27 & 0.64 & 0 & 0.71 \end{pmatrix} \begin{pmatrix} 1.69 & 0 & 0 & 0 & 0 \\ 0 & 1.12 & 0 & 0 & 0 \\ 0 & 0 & 0.84 & 0 & 0 \\ 0 & 0 & 0 & 0.42 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0.49 & -0.03 & 0.57 & 0.66 & 0 \\ 0.44 & -0.36 & -0.62 & 0.19 & -0.5 \\ 0.31 & 0.75 & 0.1 & -0.27 & -0.5 \\ 0.44 & -0.47 & 0.36 & -0.67 & 0 \\ 0.53 & 0.28 & -0.37 & -0.06 & 0.71 \end{pmatrix}$$

We can see that Σ has four non zero singular values as rank(A) = 4. Performing rank reduction by restricting our singular vectors and values , we get a rank two approximation A_3 s and a rank 2 approximation A_2 such that:

$$\frac{\|A - A_3\|_F}{\|A\|_F} = 0.18 \text{ and } \frac{\|A - A_2\|_F}{\|A\|_F} = 0.42$$

Since there is a huge jump in the percentage change from rank three ot two, we can conclude that our rank three approximation is the best representation of the database.

6 Query matching

To compare query vector q and document j, we compare angle between q and Ae_j:

$$\begin{gathered}
\cos\theta_j = \frac{(A_k e_j)^T q}{\|A_k e_j\|_2 \|q\|_2} = \frac{(U_k \Sigma_k V_k^T e_j)^T q}{\|U_k \Sigma_k V_k^T e_j\|_2 \|q\|_2} = \frac{e_j^T V_k \Sigma_k (U_k^T q)}{\|\Sigma_k V_k^T e_j\|_2 \|q\|_2}\ \forall j = 1, \cdots, d \\
\text{Define } s_j = \Sigma_k V_k^T e_j\ \text{,then } \cos\theta_j = \frac{s_j^T (U_k^T q)}{\|s_j\|_2 \|q\|_2} \forall j = 1, \cdots, d
\end{gathered}$$

Note here since $s_j = \Sigma_k V_k e_j^T$, $U_k s_j = A_k e_j^T$ and hence s_j are coordinates of $j^t h$ column of A_k in the basis defined by the columns of U_k.
Now $UU^T = I$ as U is $t \times t$ orthogonal.Hence

$$\begin{gathered}
q = I_q = [U_k U_k^{\perp}][U_k U_k^{\perp}]^T q \\
= U_k U_k^T q + U_k^{\perp} U_k^{\perp}{}^T q \\
= q_k + q_k^{\perp}
\end{gathered}$$

Here q_k is the orthogonal projection of q into column space of A_k and $U_k^{-1}q_k = U_k^T q$: coordinates of this projection in the basis defined by the columns of U_k. We can therefore use $U_K^T q$ in the cosine formula as $q_k^{\perp}$ cannot contain any information related to A_k. Hence

$$\cos\theta'_j = \frac{s_j^T(U_k^T q)}{\|s_j\|_2\|U_k^T q\|_2}$$
$$\geq \cos\theta_j \ \forall j = 1, \cdots, d$$

This cosine can be used to improve recall at the expense of precision. Note that we only have to compute s_j once for all queries. We need not calculate the full SVD of A as it is enough to compute only the first k singular values and vectors for Σ_k, U_k and V_k. Hence lowering the rank also reduces the cost of query matching.

Querying for documents related to vaping gave the following results:

Cosine of A_3 without projecting q : $\begin{pmatrix}0.45 & 0.01 & 0.99 & -0.03 & 0.70\end{pmatrix}$
Cosine of A_2 approximation after projecting q : $\begin{pmatrix}0.46 & 0.011 - 0.030.71\end{pmatrix}$

Although the first document was not retrieved, the cosine of angle increased when compared to QR. Using A_2 may return bizarre results : while trying to retrieve documents based on cigarettes and the effects on lungs, the first document was not returned but the second document about smoking lambs was returned.

7 Term-Term Comparison

Term-term comparison, implemented as part of a search engine, allows for the comparison of terms with terms. It is performed by computing the cosine of the angles ω_{ij} between all pairs of term vectors i and j using the formula:

$$\cos\omega_{ij} = \frac{(e_i^T G)(G^T e_j)}{(\|G^T e_i\| \cdot \|G^T e_j\|)}$$

where e_i denotes the i-th canonical vector of dimension t (the i-th column of the $t \times t$ identity matrix). The cosines are listed in the matrix C, where $C_{ij} = \cos\omega_{ij}$.

The cosine similarity measure used in term-term comparison provides an intuition of how closely related two terms are based on their respective term vectors.The cosine of the angle between two term vectors ranges from -1 to 1, with 1 indicating perfect similarity (term vectors pointing in the same direction), -1 indicating perfect dissimilarity (term vectors pointing in exactly opposite directions), and 0 indicating no similarity (term vectors orthogonal to each other).

Let us take an example:

Terms:

T1. Quality

T2. Efficiency

T3. Produce(ing,uction)

T4. Maximizing

T5. Art

T6. Film

Document Titles:

D1.”Efficiency and Quality Maximized: Standardizing Production Procedures for Consistent Product Excellence”

D2. .”Quality Efficiency Unleashed: Unlocking Production Success”

D3. ”Maximizing Productivity in the Workplace: Strategies for Producing More with Less”

D3. ”Maximizing Productivity in the Workplace: Strategies for Producing More with Less”

D4. ”Behind the Scenes: The Art of Film Production”

D5. ”Creating Cinematic Art: The Intersection of Film Production and Creative Expression”

The 6 × 5 term-by-document matrix before normalization, where the element a_{ij} is the number of times term i appears in document title j

$$A = \begin{bmatrix} 1.0 & 1.0 & 0 & 0 & 0 \\ 1.0 & 1.0 & 0 & 0 & 0 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 1.0 \\ 0 & 0 & 0 & 1.0 & 1.0 \end{bmatrix}$$

The 6 × 5 term-by-document matrix with unit columns:

$$G = \begin{bmatrix} 0.7071 & 0.7071 & 0.4472 & 0.7071 & 0 & 0 \\ 0.7071 & 0.7071 & 0.4472 & 0 & 0 & 0 \\ 0 & 0 & 0.4472 & 0.7071 & 0 & 0 \\ 0 & 0 & 0.4472 & 0 & 0.7071 & 0.7071 \\ 0 & 0 & 0.4472 & 0 & 0.7071 & 0.7071 \end{bmatrix}$$

The Cosines of Angles between the two vectors:

$$C = \begin{bmatrix} 1 & 1 & 0.6325 & 0.5 & 0 & 0 \\ 0 & 1 & 0.6325 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0.6325 & 0.6325 & 0.6325 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Based on the terms like ”maximizing”, ”quality”, and ”efficiency” are related to the documents of industry produce while the term ”film” and ”art” is related to the film production. This grouping of terms into semantically related groups is known as clustering. Clustering can be used as a mechanism for dealing with polysemy, where terms have multiple meanings. An automated indexing tool can use this clustering information to prompt users to identify which topic of interest, such as such as film production or industrial productivity, to refine their search.

The code for all the results derived in this project can be found here: link

8 Conclusion

- The use of fundamental mathematical concepts from linear algebra has proven to be valuable in managing and indexing large text collections.
- These techniques provide efficient and effective ways to organize, search, and retrieve information from vast amounts of digital content.
- The development of vector space models and orthogonal factorizations has greatly enhanced information retrieval technologies, allowing for improved accuracy and efficiency in dealing with large-scale text collections.
- In conclusion, the application of vector space models and orthogonal factorizations from linear algebra has greatly advanced the field of information retrieval in the context of digital libraries and the Internet.
- These mathematical concepts have provided effective means for managing, indexing, and retrieving large text collections, contributing to the transformation of how information is processed, stored, and retrieved in the digital era.

9 Limitations

- **Lack of Semantic Understanding:** Vector space models do not capture the semantic meaning of terms, leading to limitations in accurately capturing context and nuances of documents.
- **Curse of Dimensionality:** High-dimensionality of vector spaces can degrade performance and efficiency of retrieval algorithms, requiring additional computational resources.
- **Inability to Handle Dynamic Information:** Vector space models may not be well-suited for handling dynamic or changing information, requiring re-computation of vectors.
- **Sensitivity to Term Weights and Similarity Measures:** Performance of vector space models can vary based on weighting schemes and similarity measures, making selection challenging.
- **Limited Handling of Polysemy and Synonymy:** Vector space models may struggle with terms having multiple meanings or synonyms, leading to inaccurate retrieval results.
- **Lack of Contextual Information:** Vector space models do not capture temporal, spatial, or domain-specific context, limiting relevance and accuracy in certain contexts or domains.

10 Current state

Information retrieval has seen significant advancements in recent years driven by the explosion of digital data and the need to maintain and analyze petabytes of information. Machine learning and deep learning techniques are used to return personalized query results. Several Natural Language Processing (NLP) techniques like Named Entity Recognition (NER), Parts of Speech Tagging (POS), sentiment analysis and topic modelling are used to understand the semantics of queries and documents.

11 Acknowledgement

We would like to thank our professor Priyavrat Deshpande for all the guidance and support he provided us during the course of our project. We would also like to thank the authors Michael W. Berry, Zlatko Drmac and Elizabeth R. Jessup for their paper [1] which was the main reference for our project.

12 References

1 Berry, Michael W. Matrices, Vector Spaces, and Information Retrieval.

2 S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, Indexing by latent semantic analysis, J. American Society for Information Science, 41 (1990), pp. 391–407.

3 G. Kowalski, Information Retrieval Systems: Theory and Implementation, Kluwer Academic Publishers, Boston, 1997