LAB 4 – LSI and Probabilistic Models

Objective

In this Lab, you are requested to implement and compare different Information Retrieval models, namely:

1. Latent Semantic Indexation (LSI)
2. Classic BIR – Without learning data -Smooth formula
3. Classic BIR – With learning data -Smooth formula
4. Extended BIR – Without learning data -Smooth formula
5. Extended BIR – With learning data -Smooth formula
6. BM25

The goal is to understand how each model represents documents, interprets user queries, and computes relevance between them.

1. Prerequisites

Before starting this lab, ensure that you have successfully completed **Lab 1**, where you:

- Indexed the document collection $\mathbf{D_1}$–$\mathbf{D_6}$.
- Tokenized the text using the specified regular expression.
- Removed stop words and applied **Porter stemming**.
- Calculated **TF** and **TF–IDF** weights following the lecture notes formula.
- Built both the **Document–Term Matrix** and the **Inverted Index** structures.

2. Implementation

Task 1 – LSI Model

Implement the **LSI model** to retrieve documents using the TF–IDF representation built in Lab 1, and test your implementation on the queries provided below.

Test Queries

Use the following queries to test your implementation:

q1: large language models for information retrieval and ranking

q2: LLM for information retrieval and Ranking

q3: query Reformulation in information retrieval

q4: ranking Documents

q5: Optimizing recommendation systems with LLMs by leveraging item metadata

Expected Steps

1- Build the TF–IDF Matrix W

$$W_{(terms \times documents)} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{m,1} & \cdots & w_{m,n} \end{bmatrix}$$

2- Apply Singular Value Decomposition (SVD), using the numpy decomposition function

U, S, VT = np.linalg.svd(W, full_matrices=False)

3- Reduce Dimensionality to K=3
4- Represent the queries as binary vectors (1 if term exists 0 else)
5- Project them into the new latent semantic space dimensions using the formula given in the lecture notes
6- Compute similarity between the queries and all documents, using the formula given in the lecture notes
7- Rank the Documents for each query in decreasing order.

Task 2 – Probabilistic Models

1- Classic BIR – Without Learning Data

Build the classic Binary Independence Retrieval (BIR) model using the smoothed formula without learning data (i.e. no relevance feedback). For each query, compute the RSV score for every document and return a ranking (best first).

Hint: Build the binary term–document matrix first

2- Classic BIR – With Learning Data

Build the classic Binary Independence Retrieval (BIR) model using the smoothed formula with learning data (given in the table below). For each query, compute the RSV score for every document and return a ranking (best first) and give a short interpretation of the result.

Query	**Relevant documents for the query**
q1	D2, D4
q2	D2, D4
q3	D4, D1
q4	D2, D1
q5	D3, D6

3- Extended BIR with and without learning data

Implement the Extended Binary Independence Retrieval (Extended BIR) model, which generalizes the Classic BIR by incorporating term frequency information rather than treating documents as purely binary.

You will:

- Use TF–IDF values as document term weights,
- Use Binary query vectors (1 if term appears in query, 0 otherwise),
- Use Same probabilistic weighting formulas given in lecture notes
- Implement 2 models : Extended BIR With and without learning data.

4- BM25

Implement the BM25 ranking model — a modern probabilistic variant that is based on the 2-Poisson hypothesis, it balances term frequency and document length normalization using the formula given in the lecture notes, take **K1=1.5** and **b=0.75**.