Deadline: 24/12/2025 at 00:00

LAB 5 – IR Systems' Evaluation

Objective

In this final lab, you are required to **evaluate and compare the different Information Retrieval Systems (IRS)** implemented throughout the previous labs. The comparison must be carried out using **standard evaluation metrics studied in the course**, in order to identify the **best-performing model** on a real-world dataset.

1. Team Members

- Form groups of 6 to 7 students.
- Each group must designate one group leader responsible for submission via Google Classrooms.
- By **18/12/2025**, I expect through your delegate: A list of 10 groups and 1 leader per group.
- These leaders must communicate their emails to be added to Google Classroom.
- Only one submission per group is allowed.
- At the beginning of your report, clearly indicate who contributed to which part of the work.
- If identical code is found, both groups will receive zero

2. DataSet

For this final lab, the evaluation will be performed on a **real and standard IR benchmark dataset**: **MEDLINE**. It is publicly available through the University of Glasgow's IR resources: http://ir.dcs.gla.ac.uk/resources/test_collections/medl/

The collection contains three main files:

- **MED.ALL** – A set of **1,033 textual documents**, each representing a short medical abstract.
- **MED.QRY** – A set of **30 queries** designed for evaluating retrieval systems.
- **MED.REL** – The **relevance judgments (qrels)** that specify which documents are relevant for each query.

3. Evaluation Metrics

The following metrics must be implemented and analyzed:

- Precision
- Recall

- F1-Score
- Precision–Recall Curve
- Interpolated Precision–Recall Curve
- Mean Average Precision (MAP)
- Interpolated MAP
- Precision@K (P@K) (**K = 5, 10**)
- R-Precision (**R = # of relevant documents per query (from MED.REL)**)
- Reciprocal Rank (RR)
- Discounted Cumulative Gain (**Use a cutoff : DCG@20**)
- Normalized DCG (nDCG) (**Use a cutoff : nDCG@20**)
- Gain (%)

For the gain (%), use the first 10 queries (I 1 – I 10) of the MED.QRY and the metric nDCG@20 to compare.

These metrics will allow you to compare retrieval effectiveness across the different models and rank them accordingly.

4. Indexing and Preprocessing

You must apply the same preprocessing pipeline used in previous labs to ensure fair comparison between models.

1. Index the document collection.
2. Tokenize documents using the same regular expression defined in previous labs.
3. Remove stop words.
4. Apply Porter stemming.
5. Compute term weights:
    - Term Frequency (TF)
    - TF–IDF (according to the lecture notes formula)
6. Build:
    - Document–Term Matrix
    - Inverted Index

5. Retrieval Models to Evaluate

You must **reuse the already implemented models** from previous labs and the mentioned parameters as follows :

1. Vector Space Model (**Cosine Similarity**)
2. Latent Semantic Indexing (**k=100**)
3. Classic Binary Independence Model (BIR) – without relevance data
4. Classic Binary Independence Model (BIR) – with relevance data
5. Extended BIR – without relevance data
6. Extended BIR – with relevance data
7. BM25 (**k=1.2 , b=0.75**)
8. Language Model – Maximum Likelihood Estimation (MLE)
9. Language Model – Add-1 (Laplace) smoothing
10. Language Model – Jelinek–Mercer smoothing **(λ=0.2)**
11. Language Model – Dirichlet smoothing **(μ : use formula given in previous lab)**

- ✓ For models requiring relevance information, the relevance judgments provided in the MED.REL file are used.
- ✓ For language models, all query likelihoods can be computed in the log-domain to avoid numerical underflow.
- ✓ Parameter values were selected based on standard IR literature.

6. Implementation

Task 1 – Evaluation

- Evaluate all retrieval models using the metrics listed above.

Task 2 – User Interface (UI)

Develop a simple but effective UI (Web - streamlit or desk – PyQt) that allows:

- Query selection
- Display of ranked results for a selected query
- Visualization of evaluation metrics
- Visualization of Precision–Recall curves and interpolated curves

The UI does not need to be complex but must be clear, functional, and readable.

7. Evaluation Instructions

- Run all models on these queries.
- Demonstrate the results through:
  - Tables of metrics
  - Screenshots of your UI (Example : Use first Query)
  - Graphical plots (Precision–Recall, Interpolated Precision–Recall)

All these elements must be included in the final report.

Extra Bonus (Optional)

Implement a Learning to Rank (LTR) system, with complete freedom of choice:

- Approach:
  - Pointwise
  - Pairwise
  - Listwise
- Learning model:
  - Classification
  - Regression
  - Ordinal regression

If implemented: Integrate LTR results into both, The report and the UI

8. Required Submission

You must submit a single ZIP file containing:

1. Source Code

- Clean, organized, and executable code

2. Final Report (10–20 pages)

The report must include:

- Experimental setup
- Evaluation results for the first 10 queries (I1–I10) of MED.QRY.
- Comparative analysis of models
- Graphs:
  - Precision–Recall curves
  - Interpolated Precision–Recall curves
- (Optional) Learning to Rank section

Important Notes

- **Only submissions made via Google Classroom will be considered.**
- **Any submission after the deadline will receive a zero.**
- **No exceptions are permitted.**

Good Luck everyone!