University of Aveiro, Information Retrieval, 2024/2025

Assignment 2

In this assignment, you will enhance your information retrieval system by implementing a **Neural Reranker** module. This component aims to improve the ranking quality of search results obtained from your previously implemented BM25 searcher. Through this assignment, you'll gain practical experience with neural approaches to document ranking, an important mechanism in modern information retrieval systems.

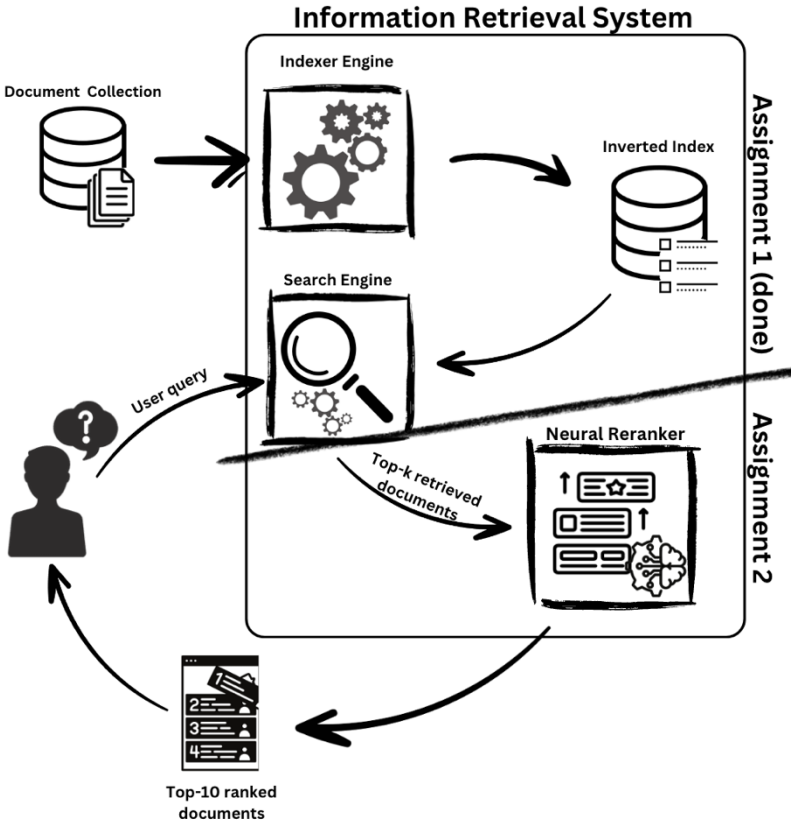

Figure 1: Overview of the components in an Information Retrieval System with a neural reranker.

The Figure above gives an overview of how the new component interacts with your current developed system.

1. Neural Reranker

Your task is to **implement** and **train** a neural reranker that processes the top-k retrieved documents from your BM25 implementation. The reranker must compute a relevance score between each document and the query, ultimately producing a refined ranking of documents. Therefore, your implementation must follow the **interaction-based** architecture, also known as cross-encoder architecture, where the model predicts relevance scores for query-document pairs. While the specific **model implementation is your choice**, you are encouraged to draw inspiration from the examples covered in the

theoretical classes. To facilitate the training process, we provide a dataset containing relevant query-document pairs specifically designed for training these types of models.

The output format of the ranking list should follow a similar structure as the question input file:
File name: final_ranked_questions.jsonl
{“query_id”:“<question_id>”,
“question”: “<question_text>”,
“retrieved_documents”: [“<doc_id_pos_1>”, “<doc_id_pos_2>”, ..., “<doc_id_pos_10>”,]},

2. Assignment Milestones

Your assignment will be evaluated based on the following key milestones:

- **Model implementation**: Demonstrate a working neural architecture that can compute relevance scores between queries and documents. This part corresponds to the implementation of the forward pass of your model.
- **Model training**: Successfully train your reranker using the provided dataset. This includes proper data preparation with negative mining, optimization, and validation of the training process.
- **Reranking Implementation**: Apply your trained model to rerank the BM25 results. Your implementation should show improved ranking quality, measured by an increase in NDCG@10 scores compared to the baseline BM25 results.

These milestones serve both as a guide for organizing your work and as evaluation checkpoints. Each milestone builds upon the previous one, helping you break down the complex task into more manageable steps.

3. Training data

To train your model we provided you a training dataset (*Dossier/Assignment2/training_data.jsonl*) with:

- 4710 unique questions with a set of relevant documents.
- Total of 40583 unique positive training pairs (question-document pair)

4. Important Details

For this assignment, you are encouraged to use your Assignment 1 BM25 implementation to retrieve the top-k documents. However, to ensure fairness across all groups, we also provide (*Dossier/Assignment 2/questions_bm25_ranked.jsonl*) a pre-computed list of top-100 documents for each query, which achieves a recall@100 of 0.8544 and nDCG@10 of 0.5253. You may use these pre-computed results if you encounter any issues with your BM25 implementation. Additionally, you can directly use these top-100 retrieved documents as input for your reranker CLI. We also provide a similar top-100 BM25 ranking for the training questions to help you with negative mining in cases where your searcher is too slow to run over the nearly 5,000 questions.

5. Submission Requirements

Your submission should include:

- README file details what was implemented and the possible configurations for executing the system.
- Well-documented source code.
- The obtained ranked lists for the 100 evaluation questions. (called it final_ranked_questions.jsonl)
- The code needs to be **reproducible**! (We recommend cloning your repo and testing reproduction before submitting). Additionally, do not forget to add a **setup script** that installs and configures all dependencies. Also, include an **entry point script** with examples of how to run your reranker.
- **Do not forget to upload your model trained weights so that we can run it.**

6. Evaluation Criteria

Your assignment will be evaluated based on the following criteria:

- Work progression along the time
- Presentations
- Information retrieval components
    1. Implementation of the neural reranker model.
    2. Training implementation.
    3. Reranking implementation.
    4. Final ranking metrics (nDCG@10).