Information Retrieval: **Optional** Assignment 3
Retrieval Augmented Generation

Prof. Toon Calders
toon.calders@uantwerpen.be

Deadline: 30/01/2026

This assignment is *optional*. If you chose not to do it, the weight of the written exam will be 66% instead of 50%.

This project is to be executed in groups of up to 4 students. These groups may, but do not have to be the same groups as for the first assignment.

1 Assignment

In the third assignment you will develop a small-scale RAG system. The project can be implemented in the programming language of your choice. The focus is on retrieval performance in terms of accuracy of the results. Runtime will only be taken into account in terms of scalability. That is: is the system architecture realistic, or will it only work for very small workloads?

You are allowed to use GenAI for help, inspiration, or even code generation. **However**, you are responsible for every line of code in your program which implies that you are supposed to fully understand the program you deliver as otherwise you could not have checked its correctness. Also, make sure that you are implementing the techniques that are taught during the course.

2 Deliverables and How to Submit

Please submit on BlackBoard a report including:

1. The names and student numbers of all group members.
2. A link to a github repository, either public, or shared with `tcalders`. This repository should include all code and data needed to run your assignment.
3. A description of the RAG system you developed.
4. (optionally) some notes you would like me to take into account when assessing your assignment.

The suggested length for the report is 4 to 6 pages.

3 Dataset - Suggestion

For this project, you will need a set of documents. There is one document collection made available as a suggestion. This document collection consists of scraped webpages of the Computer Science Masters program at the University of Antwerp, including all course descriptions.

Alternatively you could chose to use your own dataset. Your dataset should contain at least 150–200 passages after chunking.

4 System Requirements

Your system must implement the following components.

4.1 Document Chunking (10%)

- Split the dataset into passages (e.g., 100–300 words).
- Explain the chosen chunk size and any preprocessing steps[1] (e.g., lowercasing, cleaning, removing markup).

4.2 Embedding & Indexing (20%)

Use a pretrained sentence embedding model. It is up to you to select a suitable sentence embedder. Shortly motivate your choice. The sentence embedder should be able to run locally on your machine.
Tasks:

- Compute embeddings for all passages.
- Build a similarity index (e.g., cosine similarity, FAISS, or sklearn). Use an existing solution for the indexing (cfr. last assignment).

4.3 Retrieval Module (20%)

Implement the following:

- Encode the query into an embedding.
- Retrieve the top-k most similar passages. Make sure to set k to a reasonable value for your collection. (1 is too small, 20 likely too big for the suggested data collection.)
- Rank results using similarity scores.

4.4 Answer Generation Module (30%)

Use GPT-4o (via the provided API key) to generate answers.

- Accept a natural language query from the user.
- Retrieve the top-k passages.
- Insert retrieved passages into a prompt.

[1] For the provided dataset, most of the preprocessing like removing markup has already been done.

- Generate an answer using GPT-4o.
- Optionally: include retrieved passages in the output for inspection.

4.5 System Evaluation (20%)

You must evaluate your system along three dimensions:

1. **Retrieval quality**
   - Report Recall@k.
   - Manually inspect the relevance of the top-k passages.
2. **Answer quality**
   - Compare answers *with retrieval* to answers *without retrieval* (baseline).
   - Evaluate correctness, completeness, and hallucination rate.
3. **Error analysis**
   - Provide at least three cases where retrieval failed.
   - Provide at least three cases where GPT-4o produced an incorrect or hallucinated answer.
   - Explain why these errors occurred.

Evaluating RAG systems is not an easy task. This step may require some creativity from your side. You could consider schemes based on the "LLM as a judge" paradigm. Make sure to reflect on the validity of your validation.

A Note on Plagiarism

There is absolutely nothing wrong with using existing materials, you will even be commended for not reinventing the wheel, as long as you are not violating the copyright of other authors. Nevertheless, it is expected from you to clearly indicate whenever you used material that was not created by yourself. Clearly indicate in your submissions which parts constitute original work, which parts are taken from other works, and which parts were adapted from external sources. These sources have to be properly acknowledged in all your submissions. Concretely, this means at least the following guidelines are observed:

- Papers, books, webpages, blogs, etc. that were inspected while making the assignment will be referenced in a separate section "References". Citations to these materials are included in the text where appropriate.
- Text fragments exceeding one sentence that are copied from other sources are clearly marked as such. You could for instance include quoted text, definitions, etc. in italics, followed by a reference. An example of how to do this: Bela Gip (2014) defines plagiarism as *"The use of ideas, concepts, words, or structures without appropriately acknowledging the source to benefit in a setting where originality is expected"*

  References: (at the end of the document) Gipp, Bela. Citation-based plagiarism detection. Springer Vieweg, Wiesbaden, 2014. 57-88.

- When using code from other sources, indicate so in the report, and in the source code. This could for instance be done by adding a comment with a reference to the source of the function for each function that was copied from another source. It is recommended to include a separate folder "sources" in your GitHub repository with the original files from other authors that you used. Include source in the message of your commits.