Natural Language Processing - Mini-Challenge 1

Retrieval Augmented Generation

Daniel Perruchoud & George Rowlands, Institut für Data Science

1 Mini-Challenge content & learning objectives

1.1 Content

The content of the mini-challenge is the development and evaluation of a Retrieval-augmented generation (RAG) system.

Retrieval-augmented generation (RAG) is a technique that helps a Large Language Model (LLM) provide more reliable and accurate responses, and include knowledge that was not in the LLM's training set. RAG allows retrieving relevant facts from external sources beyond the LLM training data without the need for extensive retraining.

RAG works as a dialogue system that processes user queries and returns answers together with the reference to the documents passage the relevant information was retrieved from. The conversational answer is generated via LLM combining generic language knowledge and specific information chunks relevant to the query presented by the users.

In this mini-challenge you are going to implement and evaluate a RAG system. Your task is to first implement the RAG's ingestion, retriever and generator module. You then are evaluating the performance of your RAG system with the evaluator module using both human gold-standards as well as LLMs. This also includes evaluation of the groundedness of the answers by document reference.

As part of the mini-challenge you will analyze the impact of the choice of your RAG components within the main modules on the performance and quality of RAG answers. Components to be analyzed may include the chunking strategy, the vector database, the embedding model, the reranking strategy, the LLM or the evaluation framework and metrics.

1.2 Learning objectives

- LO1 - Preparation and representation of text data: ingesting cleansing and chunking of textual data, creating embeddings for textual data, retrieving information chunks matching a given query.
- LO2 - Statistical and neural language models: applying and comparing pre-trained neural LLMs to generate and automatically evaluate conversational answers for given user queries
- LO3 Transformer-based model algorithms: understanding transformed-based models used for chunk embedding and answer generation and evaluation
- LO4 Learning methods for NLP: assessing the impact of different pre-trained models and prompting strategies in the answer generation and evaluation process
- LO5 NLP Tools & Frameworks: building a RAG-pipeline with ingestion, retrieval, generator, evaluator modules and setting up and running systematic experiments

Note, that evaluation is a central part of every data science project and is particularly challenging here including different aspects:

- plausibility checks on hand-picked examples
- metric assessment of embedding results on artificially created chunk-query pairs (intrinsic evaluation)
- metric assessment of query results on human gold-standard data (extrinsic evaluation)
- automated evaluation of query results by use of LLMs (extrinsic evaluation)

2 NPR achievement of competences

In the “Natural Language Processing (NPR)” module, competence is acquired by demonstrating practical and theoretical skills.

Practical competence is assessed by means of **two mini-challenges**, **theoretical competence** by means of an **oral MSP**, which is completed after submission of the mini-challenges.

The overall assessment is made up of the graded mini-challenges (50%) and the oral MSP (50%).

3 Mini-challenge Basics

3.1 Software

Python will be used in this mini-challenge; a consistent usage of existing frameworks is to be combined with implementation of own functions.

3.2 Data

The data is made available through www.anacode.de and encompasses a corpus of around 10’000 cleantech media articles.

In addition, a small collection of human gold-standard query-passage-answer triplets will be provided for evaluation. The cleantech media corpus for this mini-challenge is accessible via Kaggle (https://www.kaggle.com/datasets/jannalipenkova/cleantech-media-dataset).

3.3 Infrastructure

The RAG solution will use resources for ingestion, chunking, storing and matching query and chunks. For answer generation and evaluation, LLM access will be available through a dedicated MS Azure **OpenAI-API key per group**.

4 Mini-Challenge submission conditions

4.1 Deliverables

4.1.1 Notebooks

Analyses must be submitted in the form of notebooks, whereby in addition to the **.ipynb file**, a version **rendered as .html or .pdf** must also be submitted. **All analyses** must be **carried out in one piece** before submission, the idea and execution of the **analyses** must be **described precisely**, and the **results** must be **documented and interpreted comprehensibly**.

4.1.2 Code repositories

In addition, a **well-structured and documented repository** of the final and executable codes must be made accessible including details on dependencies on additional libraries (i.e. requirements.txt or environment.yml file). Further recurring functionalities should be outsourced in script files, libraries or packages.

The title of the mini-challenge and authorship must be noted in the name. The analyses must be sent on time by e-mail to “daniel.perruchoud@fhnw.ch”.

4.2 Teamwork

The mini-challenge 1 is to be implemented as a team in **groups of three persons**.

Collaboration between groups is limited to conceptual aspects. In particular no code may be copied from other groups or from the Internet.

4.3 Tools

The use of ChatGPT or comparable AI tools is permitted. Their use must be noted in the deliverable for corresponding pieces of code and briefly assessed and discussed in a separate section at the end of the analysis (length 250-500 words).

The task for which the AI tool was used and which prompting strategy was used must be specified In addition, it should be described which prompting strategy was most successful, i.e. which contributed most a) to solving the task and b) to the acquisition of skills.

4.4 Exchange meetings and Deadline

Every team is required to set up in advance one **exchange meeting** during NPR contacts hours **at least three weeks before submission** of mini-challenge 1.

The deadline for this mini-challenge is May 2, 2025.

5 Mini-Challenge assessment criteria

Grades are awarded based on the four assessment criteria listed below with **priority #1 on Traceability** and **priority #2 on Completeness**.

5.1 Traceability

The analyses must be designed in such a way that both the

- underlying considerations,
- their implementation and
- the derived results

are comprehensible. This presupposes that the

- **notebooks and codes are well structured and commented** according to best practice standards,
- the **data pipeline** is **visually displayed and well explained**,
- **analytical results** are presented with tables and graphics and are **fully discussed**,
- **intermediary and final results** are analyzed on a random sample of observations and **critically inspected**.

5.2 Completeness

The **content** of the analyses must be **complete** in accordance with the description of the mini-challenge.

5.3 Correctness

The submitted analyses are checked for correctness of content. **Running your codes before form scratch** before submission is mandatory.

5.4 Best practice standards

Code repetition should be avoided by copying code and outsourced to functions that are **tested** and **logged** before use.