COMP64702 Transforming Text Into Meaning Coursework: Retrieval-Augmented Generation (RAG)

I. Introduction. Large language models (LLMs) have become integrated in many real-world applications. Although designed as auto-regressive models for next-token prediction problems such as text completion, advances in computational infrastructure have allowed researchers to scale up the amount of data that can be used to train them, making it possible to feed them massive amounts of texts (e.g., the entire Wikipedia and millions of other webpages). This has resulted in knowledge being baked into the models, in the form of billions of parameters. It is thus unsurprising that nowadays, a large proportion of LLM users utilise LLMs as they would a search engine or knowledge base, i.e., as a source of answers to questions, including those of a specialist nature.

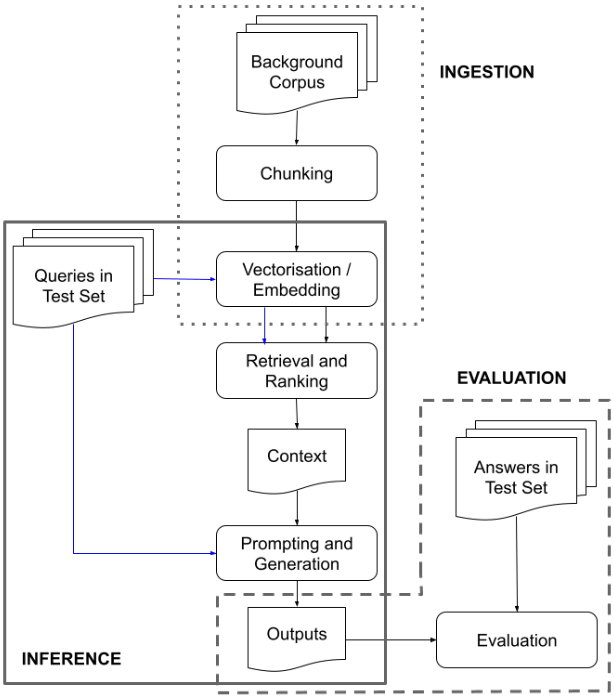
However, LLMs suffer from well-known limitations. An example is their tendency to generate *hallucinations*: sequences of tokens that convey factually incorrect information. Another is *knowledge cutoff*: the knowledge learnt by an LLM is confined within the datasets it was trained on; any information reported after the creation of those datasets is not accessible to the model. To mitigate these hallucinations, *retrieval-augmented generation (RAG)* has been integrated into many LLM-based systems, providing access to information from a curated background corpus, which the models can use as additional context during generation.

In this coursework, your self-organised group (with 3-4 members) will develop your own RAG framework.

II. Intended Learning Outcomes

- to design a bespoke approach to RAG
- to develop and evaluate custom components that comprise the bespoke approach
- to act as a responsible member of a team, communicate with team mates, and contribute to the team's self-organisation, planning and conflict resolution for the duration of the group work

III. The Problem. As part of this coursework, you will be developing your own AI-based culinary assistant, in the form of a question answering (QA) system underpinned by a RAG framework. The figure to the right depicts a typical RAG schematic, which encapsulates three pipelines:



<u>Ingestion</u>: Documents (relevant to the domain of interest) in a background corpus are pre-processed to produce chunks of text, which are then converted into embedding representations which are kept in a vector store.

<u>Inference</u>: In real time, each query (in a query set) written in the form of natural language is converted into an embedding representation. This representation is used

to retrieve the chunks of text in the background corpus that are most relevant to the given query, by searching the vector store (produced as part of Ingestion) for the most similar embeddings. When an LLM is prompted to answer the query, the highest ranked relevant chunks are provided to the LLM as additional context. The LLM then generates a response (output) for each query.

Evaluation: Using well-defined metrics, the output generated by the LLM (during Inference) is assessed against the gold standard answer to the given query.

Your culinary assistant should be specialised (i.e., should be knowledgeable) in any one of the following three cuisines:

a. East Asian cuisine: https://en.wikipedia.org/wiki/List_of_Asian_cuisines#East_Asian_cuisine
b. Mediterranean cuisine: https://en.wikipedia.org/wiki/Mediterranean_cuisine
c. South Asian cuisine: https://en.wikipedia.org/wiki/South_Asian_cuisine

IV. Coursework Phases. This coursework will be carried out in three phases. Details pertaining to deliverables, their respective deadlines and output formats are provided in succeeding sections.

A. Preparatory Phase. In this phase, you should carry out the preparatory work described below.

A.1. Together with your group mates, select the cuisine that you want to focus on. You should select only *one* cuisine.

A.2. Using *only* the publicly available sources listed below, create your own Background Corpus [Deliverable 1] and RAG Benchmark Dataset [Deliverable 2].

Wikipedia (https://en.wikipedia.org/wiki/List_of_cuisines)

Wikibooks (https://en.wikibooks.org/wiki/Cookbook:Cuisines)

Around the World in 80 Cuisines
(https://aroundtheworldin80cuisinesblog.wordpress.com/)

B. Implementation Phase. This phase is dedicated to the design and development of your RAG framework.

B.1. Design your RAG framework (see Section III) and develop *all* of the five components listed below. You are free (and encouraged to) explore or experiment with different approaches for each component, bearing in mind any constraints specified.

(1) Chunking

(2) Vectorisation/Embedding

(3) Retrieval and Ranking. Your component should return at most **5** chunks.

(4) Prompting and Generation. For the purposes of equitability, all groups are required to use the same LLM for generation: `Qwen/Qwen2.5-0.5B-Instruct`. However you are free to explore different prompting strategies.

(5) Evaluation. You can decide on the evaluation metrics that you wish to implement. We leave it to your group to do some research on the various metrics used for

evaluating key aspects of RAG systems.[1] Use the RAG Benchmark Dataset that you created in the previous phase (see Section IV A.2) to develop and evaluate any of your components.

B.2. Write and document your codebase in such a way that makes it possible -- for yourselves as well as people outside of your own team -- to run each of your Inference and Evaluation pipelines (see Section III) in real time [Deliverable 3].

C. Submission and Reporting Phase. The last phase of the coursework is focused on the presentation of the results of your work.

C.1. Prepare a 10-minute slide presentation describing how you approached: (a) the creation of your Background Corpus and RAG Benchmark Dataset; and (b) the development of your components for Chunking, Vectorisation/Embedding, Retrieval and Ranking, Prompting and Generation, and Evaluation [Deliverable 4].

C.2. Be ready to give a demonstration of each of your submitted Inference pipeline in real time. During the live demonstration, you will be given a test set (see Section IX for the format) that is unlabelled: it contains queries but not the corresponding (hidden) gold standard answers.

You will then run your Inference pipeline on these queries and produce the generated answers (test outputs) in the form of a JSON file [Deliverable 5]. Your test outputs should be written in our specified format (see Section IX), as they will be processed automatically to:

- assess the performance of your Inference pipeline based on (hidden) gold standard answers;
- compare the performance of your Inference pipeline with that of a baseline implementation that the marking team has developed.

You will also be asked to demonstrate your Evaluation pipeline. You are free to choose the input and output formats used by your Evaluation pipeline.

V. Labs. We have timetabled four labs that are relevant to this coursework.

Phase	**Lab**	**Description**	**Date**
Phase A	1	Introduction to the Coursework, Q&A	20 Feb 12:00-14:00
Phase A	2	Creating Benchmark Datasets; RAG Components	06 Mar 12:00-14:00
Phase B	3	RAG Components	20 Mar 12:00-14:00
Phase C	4	Oral Presentations and Live Demonstrations	24 Apr 12:00-14:00 (TBC)

[1] Yu et al., 2025. "Evaluation of Retrieval-Augmented Generation: A Survey" https://link.springer.com/chapter/10.1007/978-981-96-1024-2_8

VI. Deliverables, Deadlines and Timetabled Activities

Deliverable Number	Description	Format	Submission Deadline	Mode of Submission/Delivery
1	Background Corpus	Flexible	27 March	Canvas upload
2	RAG Benchmark Dataset	Flexible	27 March	Canvas upload
3	Codebase	Python notebooks	27 March	Canvas upload
4	Oral Presentation	PDF/PPT	24 April	Invigilated during Lab 4
5	Test outputs based on Live Demonstration	JSON (see Section IX)	24 April	Invigilated during Lab 4

VII. Teamwork

Each team member will be given the **same mark** for the coursework. One of the Intended Learning Outcomes of this coursework is focussed on acting as a responsible team member (see Section II), hence it is each team member's duty to ensure that tasks are delegated fairly and that there are equal contributions.

VIII. Marking Rubric

The coursework is worth 30 marks overall. The table below outlines the criteria comprising the marking rubric and the distribution of marks across these criteria.

Ethical and Responsible Use of Data		
Data collection	The group used ethically acceptable practices in collecting the data for their background corpus and RAG benchmark dataset.	2
Dataset creation	The approach to creating the background corpus and RAG benchmark dataset is technically sound.	2
Design and Development of Components		
Chunking	The group explored and experimented with different techniques/strategies in developing this component.	2
Vectorisation / Embedding	The group explored and experimented with different techniques/strategies in developing this component.	2
Retrieval and Ranking	The group explored and experimented with different techniques/strategies in developing this component.	2

Prompting and Generation	The group explored and experimented with different techniques/strategies in developing this component.	2
Evaluation	The group implemented a pipeline that evaluates key aspects of their RAG framework.	2
Oral Presentation		
Presentation content	The presentation is informative and communicates sufficient details on the group's RAG framework.	2
Presentation flow	The presentation is engaging; the group was able to explain details of their implementations in a concise and effective way.	2
Q&A (Challenge Question 1)	The group was able to answer the challenge question at a satisfactory level.	2
Q&A (Challenge Question 2)	The group was able to answer the challenge question at a satisfactory level.	2
Live Demonstration		
Inference pipeline	The pipeline works out of the box.	2
Evaluation pipeline	The pipeline works out of the box.	2
System Outputs		
Competitive performance (Retrieval)	The solution obtains performance improvement that is statistically significant, in comparison with the baseline method.	2
Competitive performance (Generation)	The solution obtains performance improvement that is statistically significant, in comparison with the baseline method.	2

IX. Appendix

Input format: Assume that the input to the Inference pipeline is a JSON file following the schema exemplified in: input_payload_sample.json

Output format: The outputs of you Inference pipeline should be written to a JSON file following the schema in: output_payload_sample.json