

Implementation Assignment 3: Retrieval-Augmented Generation

CS 394E — Contemporary Topics in LLMs · KAUST · Fall 2026

Released	Tuesday 2026-10-13
Due	Tuesday 2026-10-27, 23:59 AST, on Blackboard (extended from 2026-10-20)
Starter files	rag.ipynb, kaust_student-handbook_26_27.pdf, test_subset.csv (on the course website, Assignments page)
Weight	5% of the final grade (graded out of 10 points, plus up to 2 bonus points)

What this assignment is about

You will build a **Retrieval-Augmented Generation (RAG)** system that answers questions about the *KAUST Student Handbook 2026–27*. You will split the handbook into chunks, embed them, index them with **FAISS**, retrieve the most relevant chunks for a question, and pass them to an LLM. You will then put a **router** in front of it and run a small **parameter study** to understand how chunking and retrieval settings trade answering correctly against abstaining correctly.

This connects to the week 7 lecture and the RAG reading, and to week 5’s material on hallucination and abstention: half of the test questions have no answer in the handbook, and the right response to them is “*I don’t know*”.

What you will do

Part	What	Points
1	Build the RAG chain: FAISS vector store, document ingestion, retriever, chain	4
2	Dynamic routing: a classifier sends KAUST questions to RAG and everything else to a plain LLM	2

Part	What	Points
3	Parameter study (chunk size, overlap, top-k, similarity threshold) and report	4
Bonus	Prompt engineering that improves abstention without hurting answers	up to +2

The bonus is optional and worth up to 2 extra points.

Part 3 in brief

test_subset.csv has 20 questions: 10 whose answer is in the handbook (d1) and 10 whose answer is not (d2). Test **at least 3 values of each parameter**. You do **not** need every combination: vary one parameter at a time around a default configuration, then add a small grid for the pair of parameters you think interact. Each configuration takes 20 LLM calls, roughly 25 seconds on Mistral or 1–2 minutes on Ibex, so a well-designed study runs in about an hour at most.

Setup

- **Python 3.10 or newer** (3.12 is what we tested with) on your own laptop. No GPU is needed.
- **An LLM**, from the same two providers as assignment 2, chosen with one line at the top of the notebook (PROVIDER = "mistral" or "ibex"):
 - the **Mistral API** (free “Experiment” plan), or
 - the course’s **LLM server on Ibex** (DeepSeek V4 Flash, your personal key from the instructors). It is **best effort**: KAUST network or VPN only, and it can be slow or down when many people use it at once. It is also slower per call, so a Part 3 configuration takes 1–2 minutes instead of about 25 seconds. Don’t leave Part 3 for the last evening; if the server is down, switch to Mistral.

Reuse your key and .env file from assignment 2 (MISTRAL_API_KEY=... or IBEX_LLM_API_KEY=...). Never paste a key into the notebook.

- The first run downloads the embedding model (about 420 MB).
- No other accounts are needed: FAISS runs in memory on your machine.

What to submit

On Blackboard, submit:

1. The completed notebook rag.ipynb **with outputs**, after running it top to bottom (Kernel > Restart & Run All).
2. A **PDF report** with:
 - result tables and at least one graph for Part 3,
 - why you chose your parameter ranges,
 - explanations of the trends, supported by evidence from the logs (what was retrieved, what the model answered),
 - your answer to the question marked !!! in the notebook,
 - the bonus analysis, if you attempt it.

3. The **logs/** folder the notebook writes, zipped.

Grading

Points are shown per part in the notebook and in the table above. In Part 3 and the bonus, points go to the **quality of your chosen ranges and the clarity of your explanations**, not to the highest accuracy. Reporting 1.0 accuracy with the largest chunk size everywhere earns little; explaining *why* a setting helps the answerable questions and hurts the unanswerable ones earns a lot.

Expected effort

About **one working day**: a few hours for Parts 1–2, an hour or so of mostly unattended experiment runs, and the rest for the report.

Late policy

10% penalty per 24 hours (or fraction thereof) up to 72 hours late, after which a zero is recorded without an approved extension.

Use of AI assistants

Using AI tools for this assignment is **not forbidden, but discouraged**: the point is to build and understand the pipeline yourself. If you do use them, **disclosure is mandatory**: at the end of your report, name the tools you used and what you used them for. You remain responsible for everything you submit.