

Worksheet: RAG, Elasticsearch 101, and Agents

CS 5090 · Software Engineering with Generative AI · Week 7 · collected for completion credit. Discussing with your neighbor is encouraged; write your own answer.

Name: _____

Part 1 · Why RAG, and what it is

warm-up 1.1 A developer asks an LLM: "What does our internal function `sync_ledger()` do?" The model confidently describes a function that does not exist. Which of the problems from the lecture is this? Circle one.

- (a) knowledge cutoff (b) lack of private or proprietary data (c) wrong output format (d) model is too small

order 1.2 RAG has four core components. Number them 1–4 in the order they happen, and write one phrase saying what each does:

#	Component	What it does
	Generation	
	Ingestion	
	Augmentation	
	Retrieval	

classify 1.3 For each situation, which approach fits best: **prompt engineering**, **RAG**, or **fine-tuning**? (Think about the two axes from the lecture: does it need external knowledge? does it need changing the model?)

Situation	Approach
Answers must cite your company's policy documents, which change every month	
The model should always answer in your team's strict JSON style, no outside knowledge needed	
The model must write code in an in-house language it has never seen, and you have 50,000 examples	

Part 2 · Elasticsearch 101

work it out 2.1 Three tiny documents:
D1: fix login bug D2: login page redesign D3: bug in payment page
Fill in the **inverted index** (term → list of documents containing it):

term	documents
bug	
login	
page	
fix	

apply 2.2 Using your index, which document(s) match the query `login AND bug`? _____ Which match `page`? _____
Why is this faster than reading every document for each query?

recall 2.3 In an Elasticsearch cluster, data is split into pieces stored across machines. Match: a **shard** is _____ and a **node** is _____.

- (a) one machine in the cluster (b) one piece of an index's data (c) one user query

recall 2.4 In the "RAG in practice" diagram, the user query goes to Elasticsearch first, then to the language model. What does Elasticsearch return, and what does the language model do with it?

Part 3 · Where RAG fails

match 3.1 Match each situation to a failure point (FP1 Missing Content · FP2 Missed Top-Ranked · FP3 Context Consolidation · FP4 Extraction Failure · FP5 Wrong Format · FP6 Incorrect Specificity):

Situation	FP
The right document exists in the index but ranked 12th; only the top 5 are used	
The answer is in the retrieved text, but the model answers from a contradicting sentence nearby	
The question is about a product the documents never mention, and the model answers anyway	
The user asked for a table; the model wrote three paragraphs	

recall 3.2 DocPrompting retrieves documentation before generating code. In one sentence: what kind of coding task does this help most, and why?

Part 4 · Agent basics

order 4.1 You tell an agent: "Fix this bug in my repository." Number the steps 1–5 in the order from the lecture: ____ Act (edit the code) ____ Inspect (files and error report) ____ Adapt (retry or ask) ____ Plan (a fix) ____ Check (run the tests) What is the name of the loop that connects them? _____

label 4.2 ReAct interleaves **Thought**, **Action**, and **Observation**. Label each line:

run_tests(module="api")	
2 failed: test_upload_size, test_upload_type	
The failures are both in upload; I should look at upload.py first.	

recall 4.3 Tool calls and MCP. True or false, circle one each:

When the model "calls a tool," the model itself executes the command.	T F
An MCP server exposes tools (and prompts, resources) that an MCP client can use.	T F
With MCP, each assistant needs its own custom integration for every tool.	T F

compare 4.4 Reflexion adds something to the loop that plain ReAct does not have. What is it, and what does the agent do with it on the next attempt?

exit ticket 4.5 In the "fix this bug" example, which step tells the agent whether it is done? Why does that make software a good place for agents?