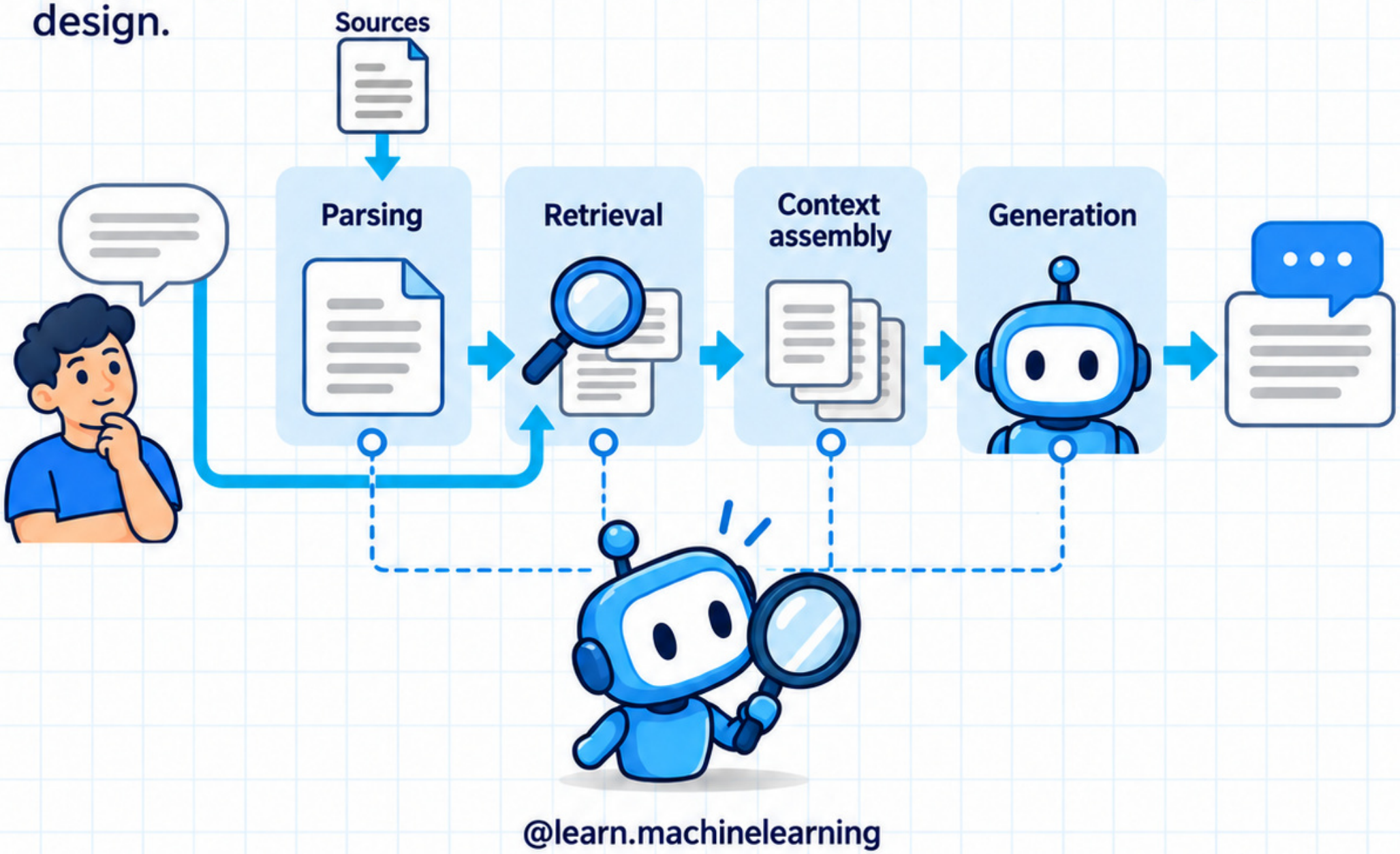
EVALUATE THE COMPLETE RAG PATH

A wrong answer can start with parsing, retrieval, context assembly or generation.

Evaluate stages separately, then measure the complete task.

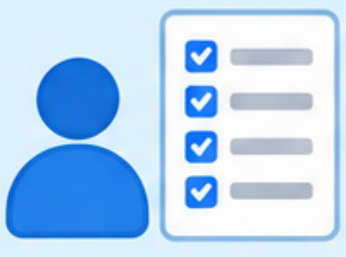
Today: diagnose a fictional return-policy assistant before changing its design.



WRITE THE TEST CASE CONTRACT

Record the question, user scope, source snapshot and expected behavior.
Label required evidence and the supported answer facts.
Include cases where the correct outcome is denial, clarification or no answer.

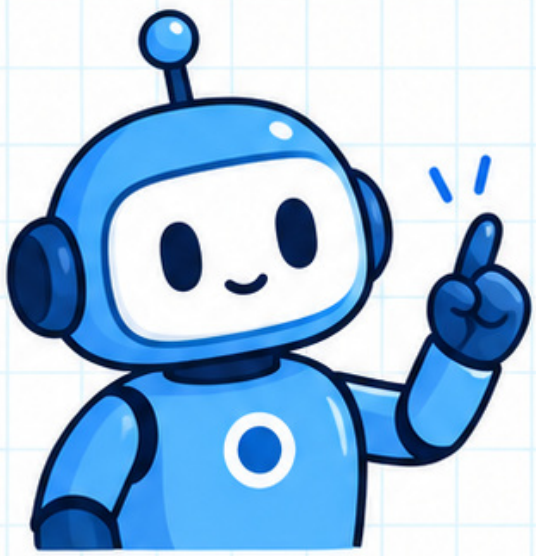
Question



User Scope

Source Snapshot

Expected Behavior



Test Case

Question:
User Scope:
Source Snapshot:
Expected Behavior:
Required Evidence:
Supported Answer Facts:

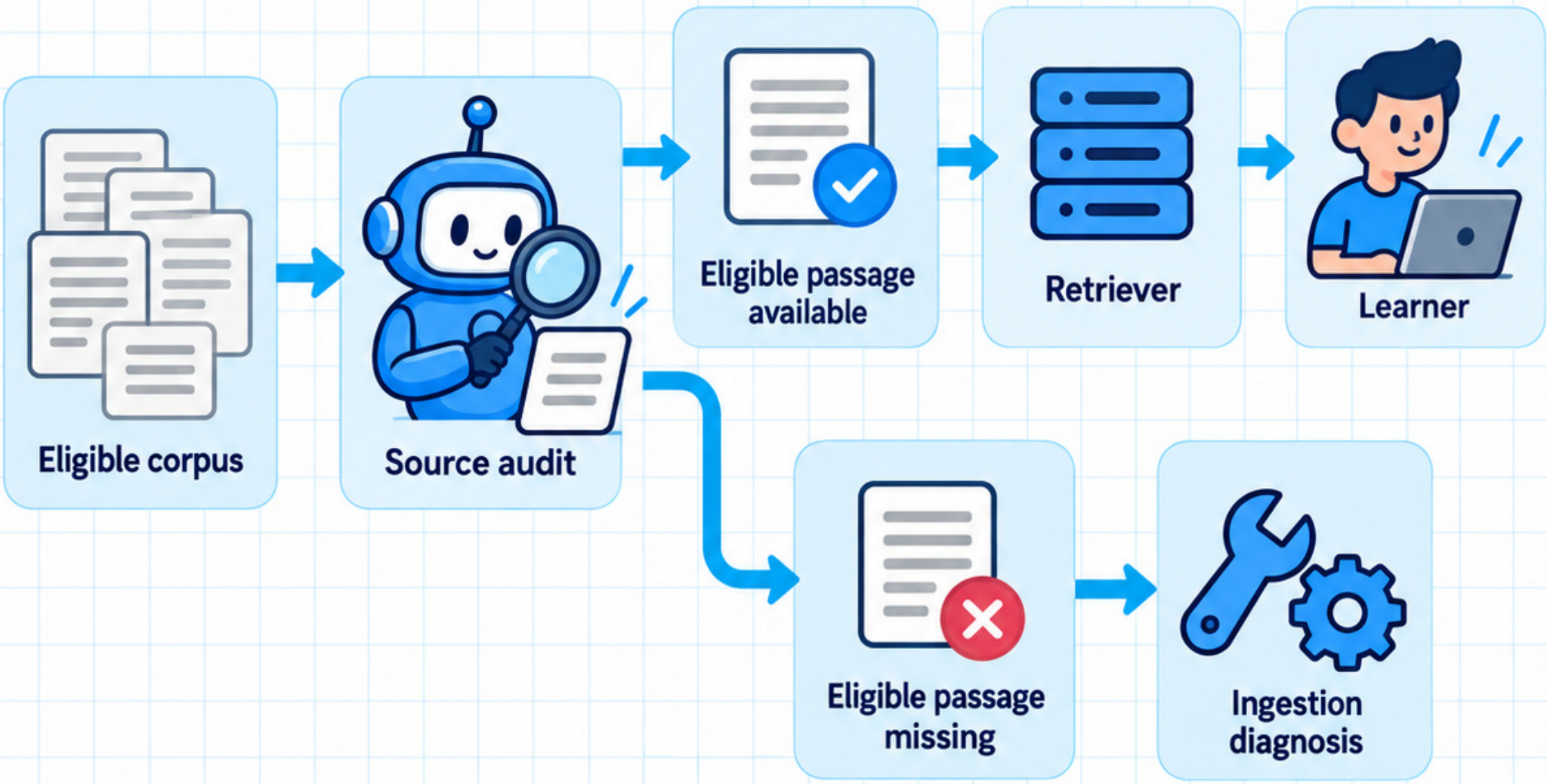
- Answer
- Denial
- Clarification
- No Answer

COVER USEFUL TASK SLICES

Include exact IDs, paraphrases, exceptions, missing facts and conflicting sources.
Add revoked access, withdrawn documents and outdated versions.
Report results by slice so an average cannot hide a critical failure.

CHECK SOURCE AVAILABILITY FIRST

Is the required passage present, correctly parsed, current and authorized?
If it is absent from the eligible corpus, retrieval cannot return it.
Fix source coverage or eligibility logic before blaming the answer model.



MEASURE EVIDENCE RETRIEVAL

Recall@K: relevant items retrieved in the top K, divided by all relevant items for the case.

$$\frac{\text{relevant items retrieved in the top K}}{\text{all relevant items for the case.}}$$

all relevant items for the case

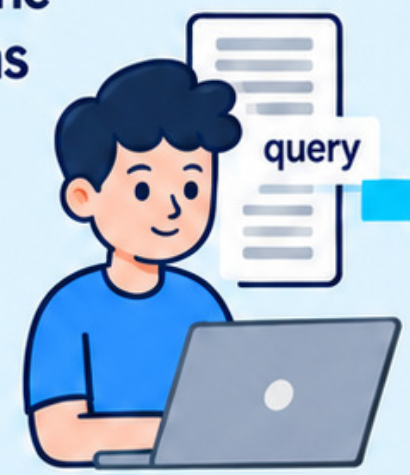
Compare sets

top K

relevant items retrieved in the top K

Precision@K: relevant items in the top K, divided by K when K items are returned.

$$\frac{\text{relevant items in the top K}}{\text{K when K items are returned.}}$$



retrieval system

top K

relevant items in the top K

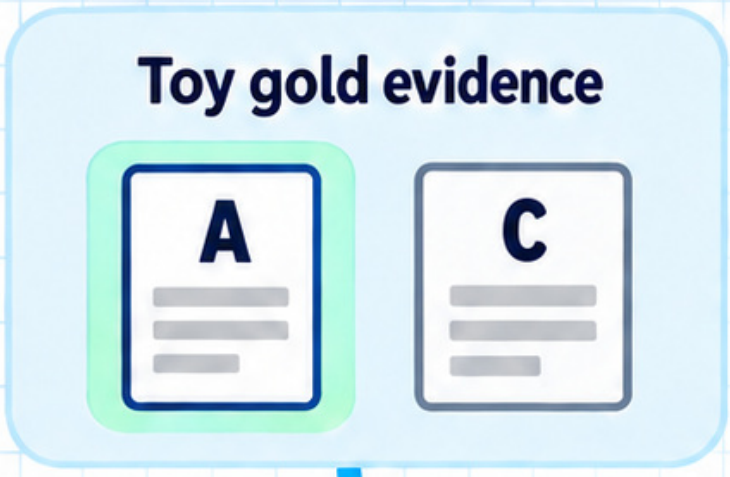
Define the relevance unit and handle fewer returned items explicitly.

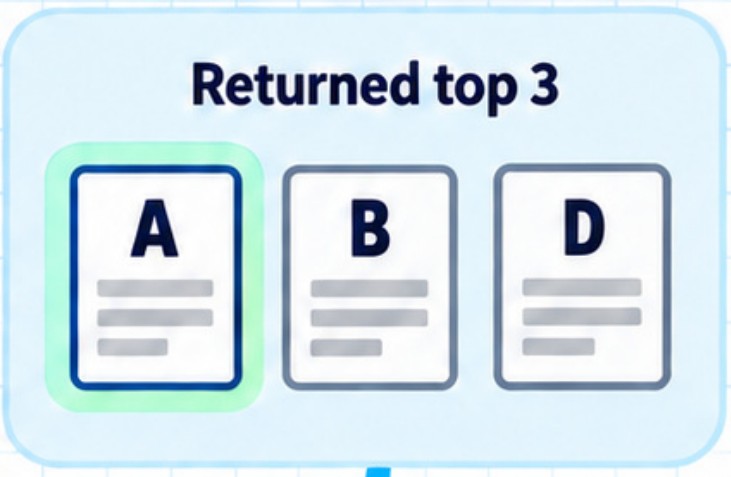
WORK THROUGH A TOY RESULT

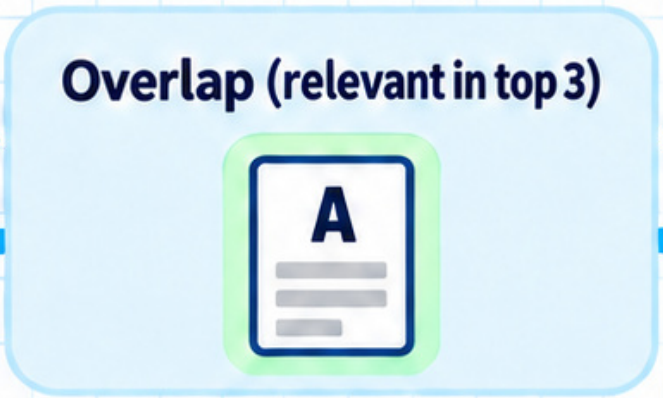
Toy gold evidence: A and C. Returned top 3: A, B, D.

Recall@3 = 1/2 = 50%. Precision@3 = 1/3 ≈ 33%.

These numbers describe this toy retrieval list, not answer correctness.







Recall@3

1/2 = 50%

Precision@3

1/3 ≈ 33%

NEAREST-NEIGHBOR RECALL DIFFERS

ANN recall compares approximate neighbors with an exact-search reference.

Evidence recall compares retrieved items with human-labeled relevant evidence.

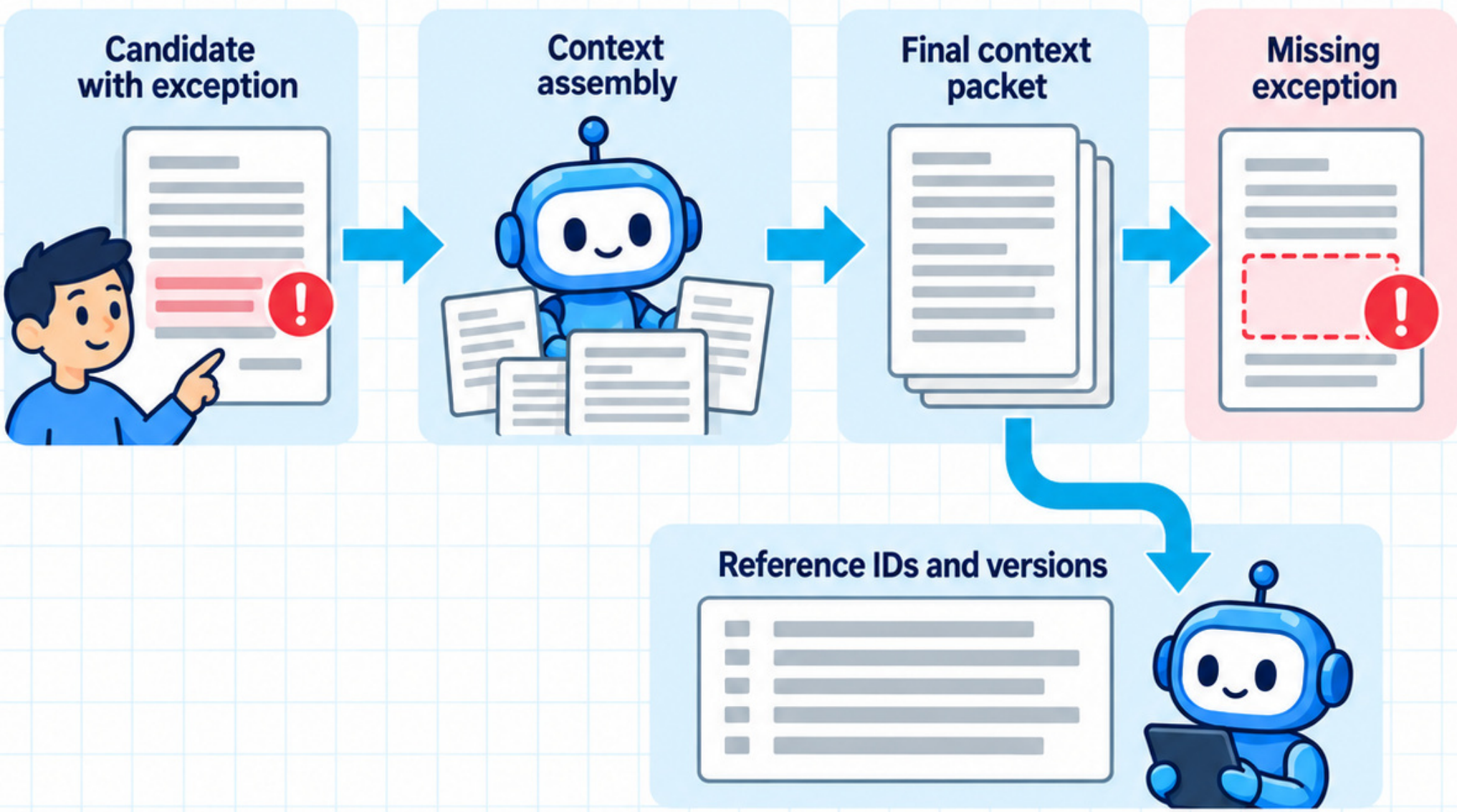
A fast, accurate neighbor search can still miss the passage the user needs.

INSPECT THE ACTUAL CONTEXT

A relevant candidate may be dropped, truncated or drowned in duplicates before generation.

Check required-fact retention in the final context packet.

Log reference IDs and versions so the loss can be located.



SCORE THE ANSWER SEPARATELY

Correctness:
does the answer meet the task requirements?

Faithfulness:
do its claims follow the supplied evidence?

Citation support:
do the linked passages support the attached claims?

These checks answer different questions.

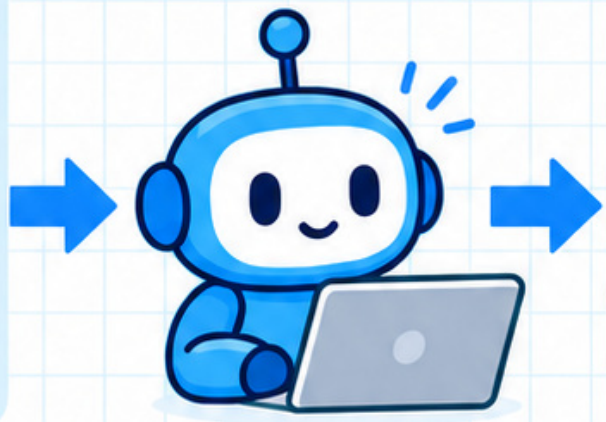
AI SYSTEM DESIGN • DAY 30

10/18

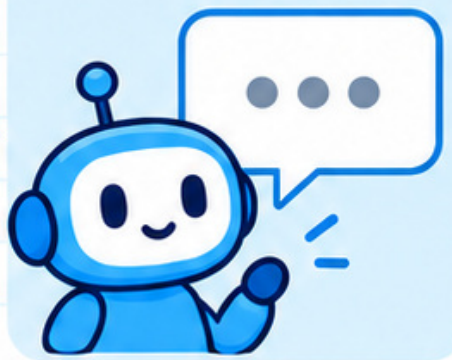
CHECK BEHAVIOR ON EMPTY EVIDENCE

A correct no-evidence response can be a successful task outcome.
Measure inappropriate answers when the allowed sources are insufficient.
Also check whether answerable cases are refused unnecessarily.

Insufficient Evidence

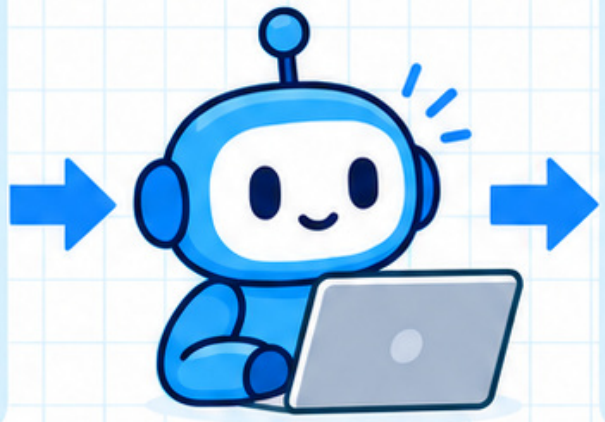


Defined Abstention



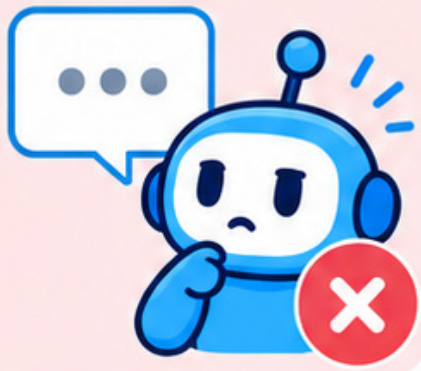
Inappropriate Answer

Sufficient Evidence



Supported Answer

Unnecessary Refusal



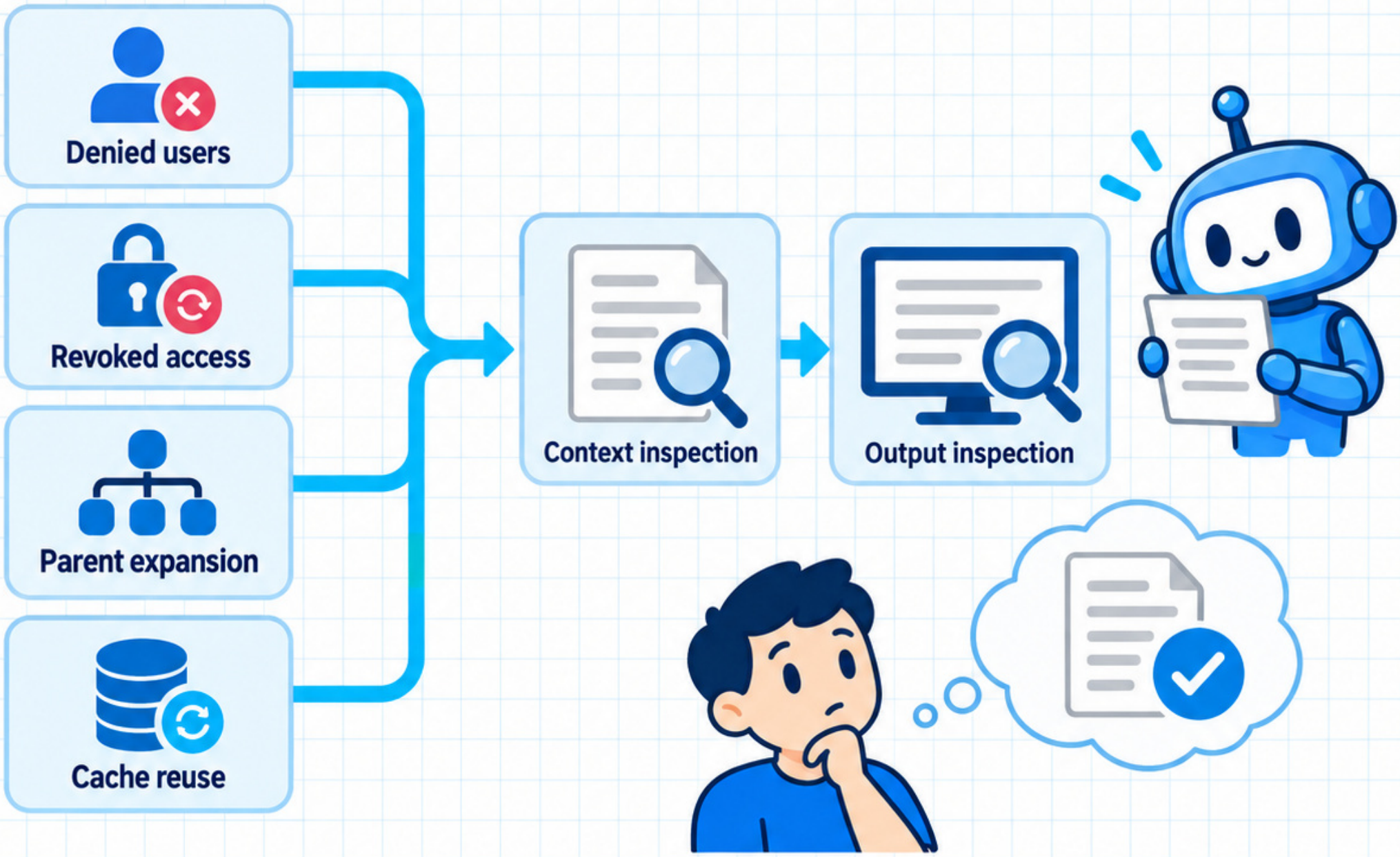
@learn.machinelearning

KEEP PERMISSION TESTS EXPLICIT

Test denied users, revoked access, parent expansion and cache reuse.

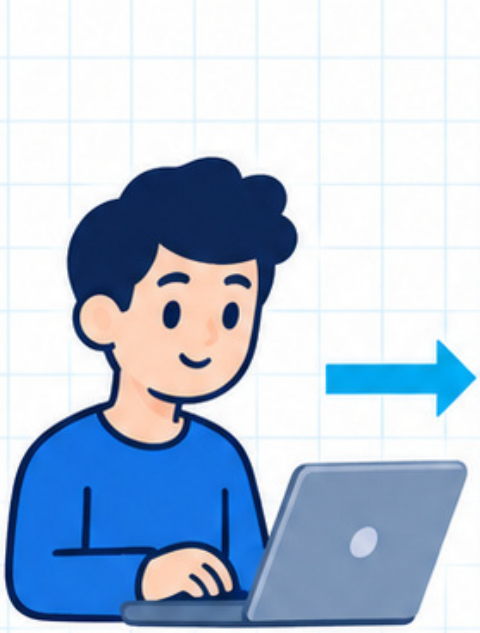
Measure protected-source exposure in context and visible output.

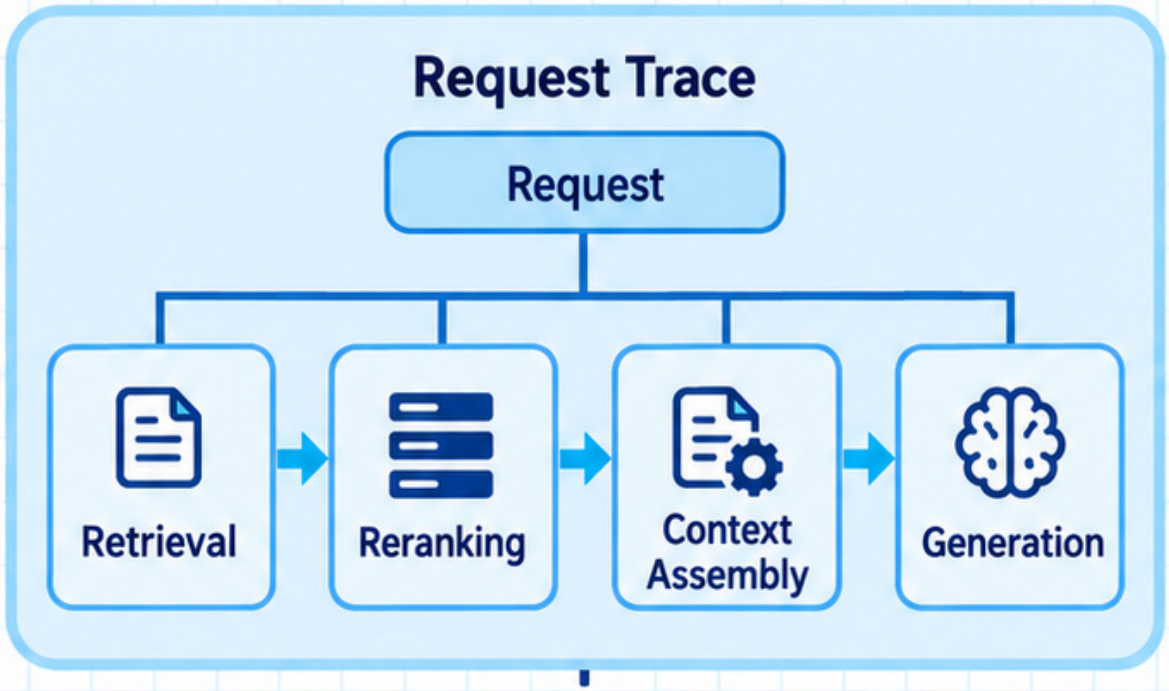
Zero failures in a sample is a release requirement for that sample, not proof of universal safety.

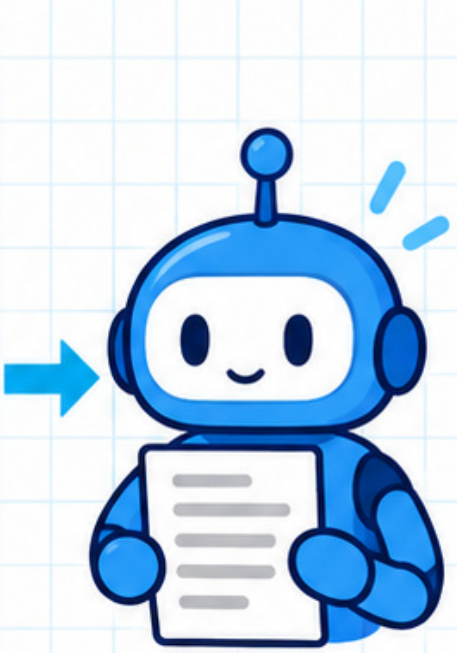


TRACE THE COMPLETE REQUEST

Use spans for retrieval, reranking, context assembly and generation.
Record stage timing, errors and version identifiers.
Traces explain what happened; they do not automatically judge whether the answer is good.





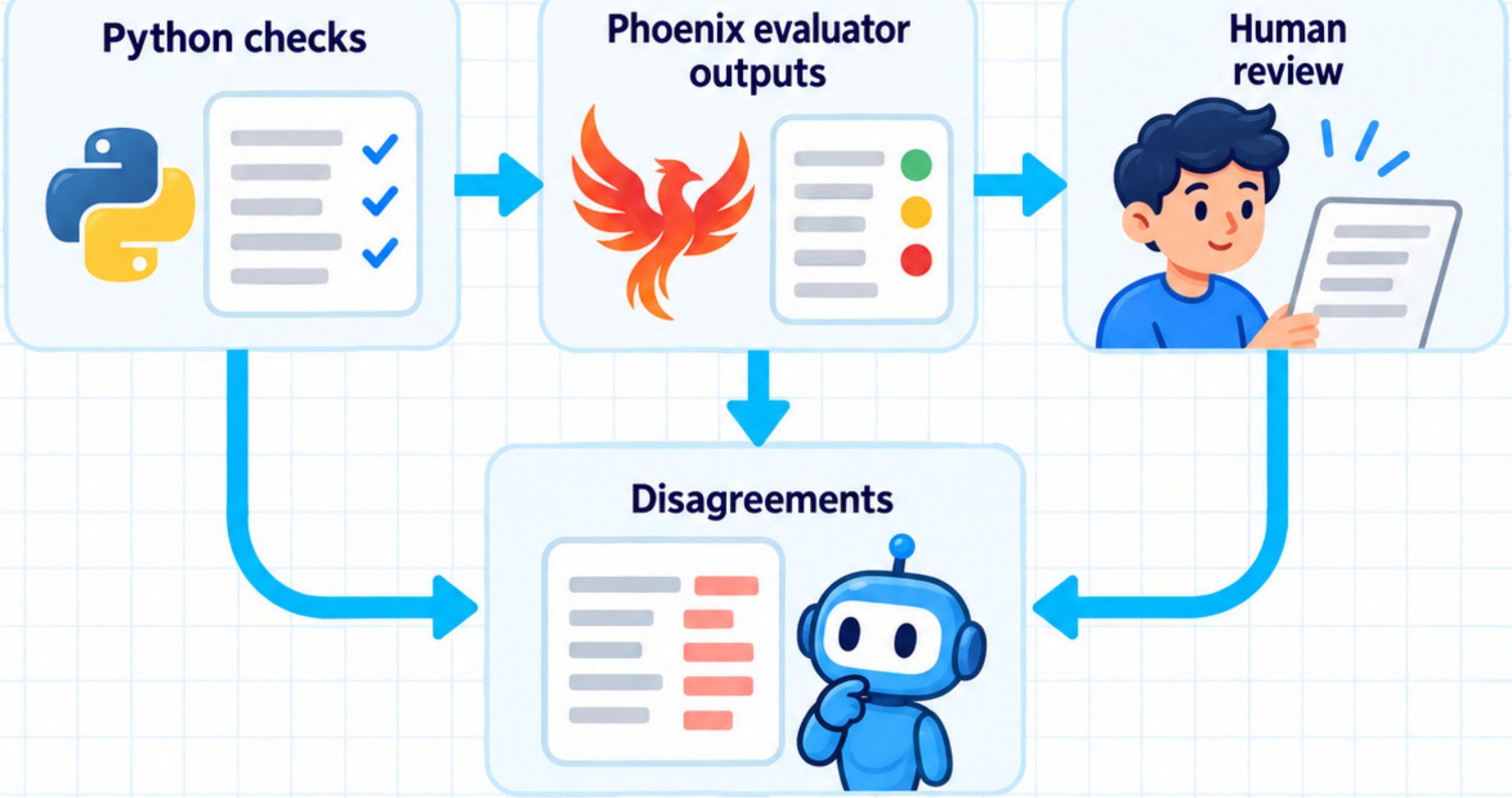


USE EVALUATORS WITH REVIEW

Python computes deterministic checks and retrieval metrics.

Phoenix helps compare runs and inspect relevance or faithfulness judgments.

Calibrate model-based judges against reviewed examples and inspect disagreements.



LOCATE THE FAILURE BEFORE FIXING IT

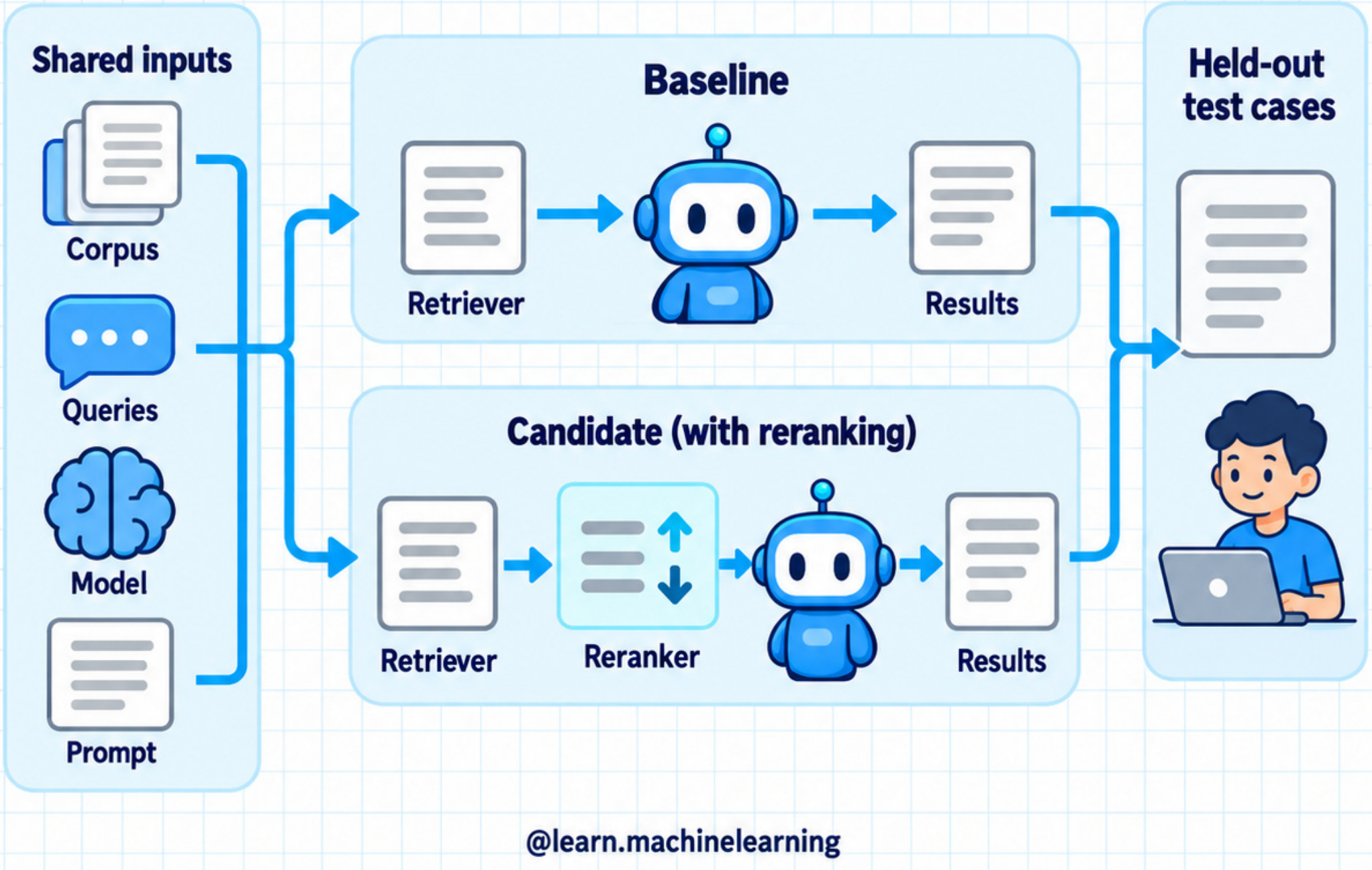
Missing source	Ingestion	inspect ingestion
Missing candidate	Retrieval	inspect retrieval
Missing context fact	Assembly	inspect assembly
Unsupported claim with good context	Generation	inspect generation
Wrong permissions	Access boundary	fix the access boundary before release

RUN ONE CONTROLLED EXPERIMENT

Toy experiment: add reranking while keeping corpus, queries, model and prompt fixed.

Use the same held-out cases and declare the success criteria first.

Report sample size and uncertainty; do not tune repeatedly on the final test set.

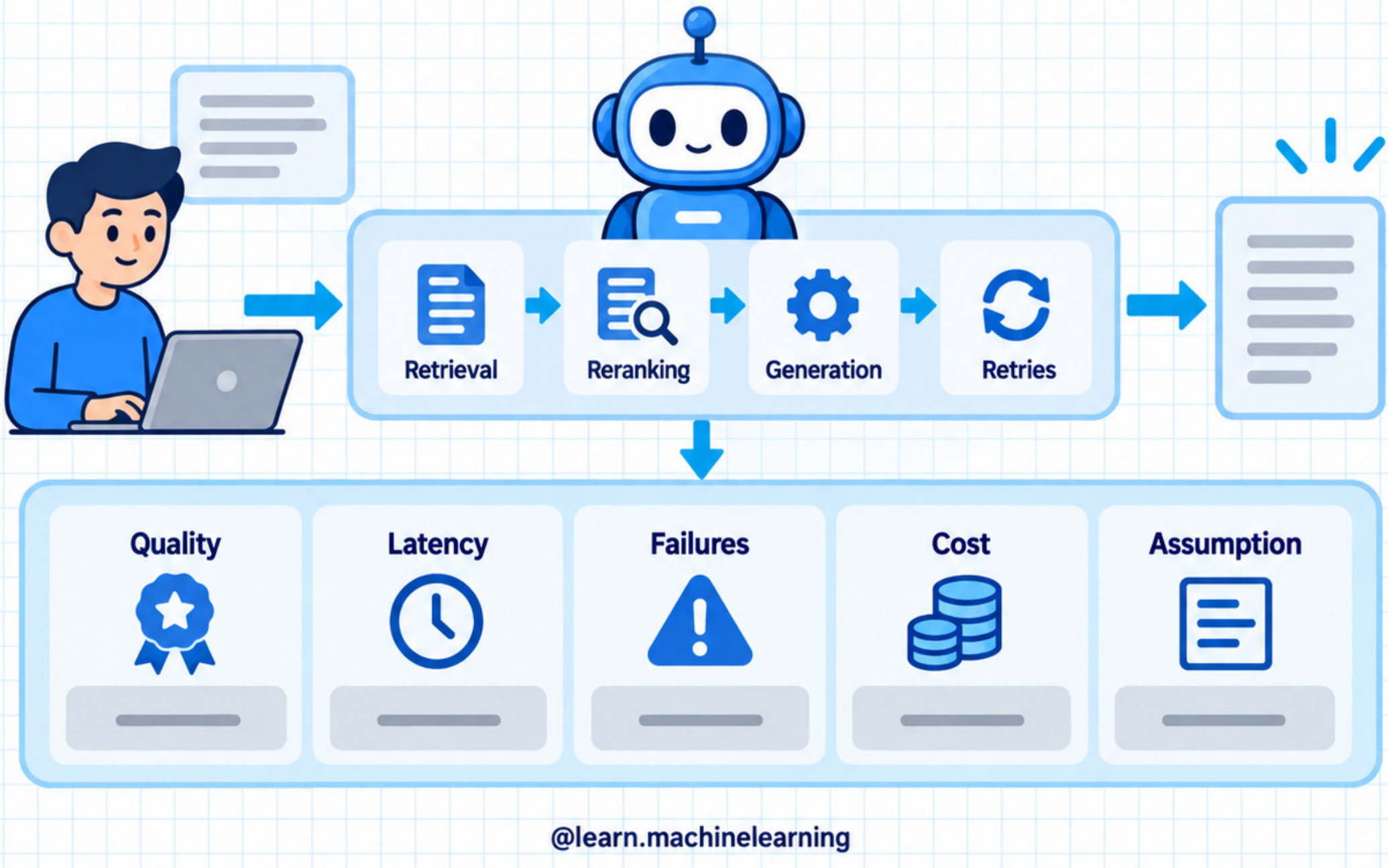


INCLUDE TIME AND COST

Report task success, p50/p95 latency and failed-request rate together.

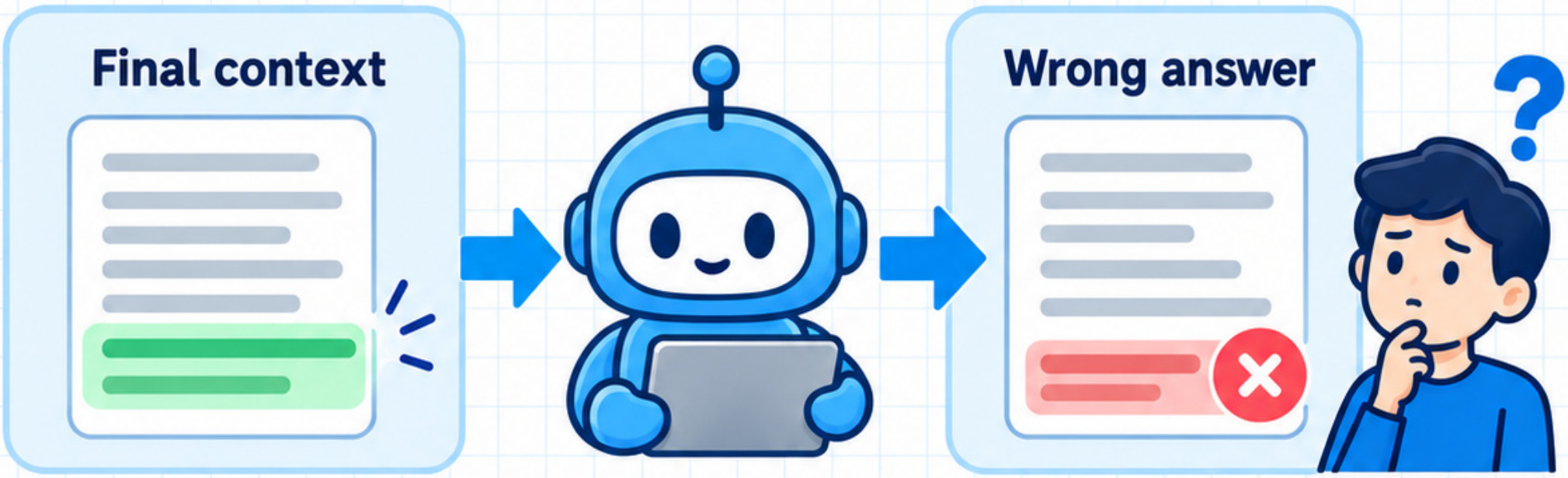
State what cost includes: retrieval, reranking, generation and retries.

A quality improvement matters only if the complete design meets its constraints.



TRY THE GOOD-CONTEXT FAILURE

Toy case: the final context contains the headphone exception.
The answer still says all opened headphones are excluded.
Which stage is the first suspect, and what controlled change would you test?



First suspect stage

Controlled change to test

AI SYSTEM DESIGN • DAY 30

18/18

YOUR COMPLETE RAG REVIEW

Audit sources, candidates, context, claims and access.
Choose changes from stage evidence, then rerun the same task suite.

You now have the RAG foundation.
Next: bounded agents and authorized tools.

@learn.machinelearning