

Evaluating RAG decisions

Precision, recall, macro-F1 and LLM-as-judge

98%

Correct disposition decisions
on 400 calibration requests

The confusion matrix explains which eight requests the system missed.

The underlying counts

Rows: expected. Columns: observed.

	Answer	Block	Redirect	Abstain
Answer	92	0	0	8
Block	0	100	0	0
Redirect	0	0	100	0
Abstain	0	0	0	100

392 / 400 = 98%

Eight answer cases became abstentions. The other 300 decisions matched their expected classes.

Precision and recall

Treat each disposition as one class.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Of the predicted answers, how many belonged to answer cases?

$$92 / 92 = 100\%$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

Of the cases that needed an answer, how many received one?

$$92 / 100 = 92\%$$

TP: correct prediction of the class. FP: incorrect prediction.
FN: missed case of the class. These counts concern dispositions.

The same errors affect abstention precision

108 predicted abstentions, 100 expected abstentions.

100 / 108

92.6% precision

The extra eight abstentions came from answer cases. Abstention recall is 100/100 because the system found every case labeled abstain.

Reviewing both classes makes the cost of unnecessary abstention visible.

Macro-F1 gives each disposition equal weight

It combines precision and recall before averaging.

$$F1 = 2 \times P \times R / (P + R)$$

Disposition	F1
Answer	0.9583
Block	1.0000
Redirect	1.0000
Abstain	0.9615

Macro-F1 $\approx$ 0.980

Unweighted mean across four classes. Accuracy also rounds to 0.980 here, but uses a different calculation.

Semantic similarity

Related wording can share an embedding direction.

$$\cos(\mathbf{q}, \mathbf{e}) = (\mathbf{q} \cdot \mathbf{e}) / (\|\mathbf{q}\| \times \|\mathbf{e}\|)$$

The rule layer compares a request with configured examples. Retrieval compares it with candidate evidence.

The score is not a policy-violation probability. Similarity also cannot establish whether a passage supports a claim.

Nonzero vectors required. Unit-normalized vectors make cosine equal to the dot product. Re-evaluate thresholds after a model change.

Runtime verification and the evaluation judge

They work at different points in the workflow.

Runtime verifier

Checks a candidate answer against retrieved text before release.

Evaluation judge

Scores saved outputs against a defined rubric and reference.

The judge does not control live answers. Agreement with a reference depends on the rubric and the quality of those labels.

79.5% exact judge agreement

159 / 200 outputs matched all five labels: grounded, privacy safe, injection safe, integrity safe and refusal appropriate.

200 outputs

Rubric selection, then lock the rubric.

200 different outputs

Validation on separate case families.

Reference: one person, two recommendation-assisted passes, then reconciliation. Independent reviewer agreement remains unmeasured.

What the grounded label counts

91.5% agreement with this combined rubric.

Expected disposition

A matching block, redirect or abstention counts as grounded.

Support for an answer

For answers, the judge also checks claims against the retrieved text.

The denominator is 200 saved outputs. It is not a set of 200 independently fact-checked answers.

Citations and human review

Different references produce different measurements.

303 / 303

Evaluated citations accepted by the runtime verifier.
This measure depends on that same verifier.

0.766

Expected-document citation precision. The difference needs independent source and label review.

The aggregate cannot tell us whether a different source is valid or a citation is wrong.

Sources and next evaluation

Keep the final system holdout separate.

Independent labels, a frozen protocol and explicit error costs are the next evaluation priorities.

Recorded project evidence

Metric implementation

Precision, recall and F1 definitions

Cosine similarity definition

Source revision: 7e886ea

The 400-case system holdout remains unopened.