Evaluation & Wrap-Up for RAG Systems

Understanding how to properly evaluate and optimize Retrieval-Augmented Generation systems for maximum performance and reliability.

Key Goals

- Understand how to measure both correctness and faithfulness of answers
- Learn common evaluation metrics used in RAG & related tasks
- See how tuning (chunk size, top-k, overlap) affects performance
- Know how to detect and limit hallucinations

🧠 Retrieval Metrics

These measure how well the retrieval component (before generation) is doing: whether it brings in relevant documents/chunks.

Metric	What it measures	Example
Precision@k	Fraction of retrieved top-k items that are relevant.	If you retrieve 5 documents for a query, and 3 of them are relevant, Precision@5 = 3/5 = 0.6.
Recall@k	Fraction of all relevant items that appear in top-k retrieved ones.	If there are 10 relevant docs overall, and your top-5 retrieval contains 4 of them, Recall@5 = 4/10 = 0.4.
Mean Reciprocal Rank (MRR)	How early (on average) the first relevant document appears in the ranking.	If for Query1 the first relevant doc appears at rank 2 (so reciprocal rank is 1/2 = 0.5), for Query2 at rank 4 (1/4 = 0.25), then MRR for those two = (0.5 + 0.25)/2 = 0.375.
Mean Average Precision (MAP@k)	Takes into account both relevance and position; average of precision at all relevant-item ranks, averaged across queries.	If a query has relevant docs at positions 1, 3, 7 out of retrieved 10, compute precision at each of those cutoff points and average.

📊 Retrieval Metrics Examples

Precision@k Example:

Query: "What is machine learning?"

Retrieved 5 documents: [ML textbook chapter, AI history, cooking recipe, ML tutorial, sports article]

Relevant documents: 3 out of 5 (ML textbook, ML tutorial, AI history)

Precision@5 = 3/5 = 0.6 (60% of retrieved docs are relevant)

Recall@k Example:

Total relevant documents in database: 10

Retrieved in top-5: 3 relevant documents

Recall@5 = 3/10 = 0.3 (30% of all relevant docs were retrieved)

MRR Example:

Query 1: First relevant doc at rank 2 → reciprocal rank = 1/2 = 0.5

Query 2: First relevant doc at rank 4 → reciprocal rank = 1/4 = 0.25

Query 3: First relevant doc at rank 1 → reciprocal rank = 1/1 = 1.0

MRR = (0.5 + 0.25 + 1.0) / 3 = 0.583

MAP@k Example:

Query with relevant docs at positions 1, 3, 5 out of top-10:

- Precision@1 = 1/1 = 1.0
- Precision@3 = 2/3 = 0.67
- Precision@5 = 3/5 = 0.6

Average Precision = (1.0 + 0.67 + 0.6) / 3 = 0.76

📝 Generation Metrics

These measure how good the generated answer is, given the retrieved context + reference (if available).

Metric	What it measures	Example
ROUGE (ROUGE-1, ROUGE-2, ROUGE-L, etc.)	Overlap between generated answer and reference text. Higher = more overlap.	If reference: "The cat sat on the mat." and generated: "The cat is sitting on a mat." ROUGE-1 (unigram) quite high, ROUGE-2 (bigrams) somewhat lower.
BLEU	Precision of n-grams in generated vs reference; penalizes missing or wrong phrases.	If generated uses many exact phrases from reference, BLEU score is high; if wording very different, BLEU drops.
BERTScore	Semantic similarity using embeddings; captures similarity beyond exact n-grams.	If generated paraphrases reference, BERTScore will reflect similarity even when ROUGE/BLEU are low.
Answer Relevancy	How much the generated answer is relevant to the question (addresses the query). Doesn't check correctness necessarily.	If question: "What causes rain?" and answer talks about evaporation & clouds, good relevancy. If it digresses into unrelated info, relevancy lower.
Faithfulness / Groundedness	Whether the generated answer's claims are supported by the retrieved documents/context; limit hallucinations.	If the answer says "X is true" but that fact is nowhere in the retrieved context, that lowers faithfulness.
Answer Correctness / Accuracy	Whether the answer is factually correct (compared to reference / trusted source) given the retrieved info.	If retrieved document says "COVID-19 first identified in 2019 in China" and generated answer says same, that's correct; if it says "2020", that's wrong.
Hallucination Rate	The proportion of statements in generated answer that are not supported by context or truth.	If answer has 5 claims, 2 are unsupported → hallucination rate ≈ 40%.

📝 Generation Metrics Examples

ROUGE Example

Reference: "The cat sat on the mat and slept peacefully."

Generated: "A cat was sitting on a mat and sleeping."

- ROUGE-1 (unigrams): 5 matching words out of 8 → 0.625
- ROUGE-2 (bigrams): 1 matching bigram ("on a/the") out of 7 → 0.14
- ROUGE-L (longest sequence): "cat...on...mat" → 0.5 Can remember it as recall

BLEU Example

Reference: "The quick brown fox jumps over the lazy dog"

Generated: "A quick brown fox leaps over a lazy dog"

- 1-gram precision: 7/9 = 0.78
- 2-gram precision: 5/8 = 0.625
- BLEU score considers all n-grams with brevity penalty
- Can remember it as precision

BERT Score Example

Reference: "The weather is sunny today"

Generated: "It's a bright day outside"

- Word-level semantic similarity using BERT embeddings
- BERTScore: 0.85 (high semantic similarity despite different words)
- ROUGE would be low due to no exact word matches

Answer Relevancy Example

Question: "How do you make coffee?"

- Good answer: "Grind beans, add hot water, brew for 4 minutes" → High relevancy
- Poor answer: "Coffee was discovered in Ethiopia centuries ago" → Low relevancy (factual but irrelevant)

Faithfulness Example

Retrieved context: "Paris is the capital of France. It has 2.2 million residents."

- Faithful answer: "Paris, France's capital, has about 2.2 million people"
- Unfaithful answer: "Paris has 3 million residents and is known for its beaches" (wrong population, unsupported claim about beaches)

Hallucination Rate Example

Answer with 4 claims: [Paris is capital ✓, Has 2.2M people ✓, Famous for Eiffel Tower ✗, Has best beaches ✗]

- Hallucination rate = 2/4 = 50% (2 unsupported claims out of 4 total)

🎯 When to Use Which Metric

For Retrieval Evaluation:

Precision@k

- You care about the quality of retrieved results
- False positives are costly (retrieving irrelevant docs wastes generation resources)
- You have limited context window for generation

Recall@k

- You want to ensure you don't miss important information
- False negatives are costly (missing relevant docs hurts answer quality)
- You have comprehensive knowledge base coverage

MRR

- You care about finding the first relevant result quickly
- Users typically look at top results only
- Speed and early relevance matter most

MAP@k

- You need balanced view of precision across all relevant items
- Multiple relevant documents exist per query
- You want to reward systems that rank relevant items higher

For Generation Evaluation:

ROUGE

- You have reference answers available
- Exact word overlap matters
- Evaluating summarization tasks

BLEU

- You have reference answers
- Precision of exact phrases is important
- Evaluating translation-like tasks

BERTScore

- You want semantic similarity beyond exact words
- Paraphrasing should be rewarded
- Reference answers use different vocabulary

Answer Relevancy

- You want to check if the answer addresses the question
- No reference answer available
- Evaluating question-answering systems

Faithfulness

- Preventing hallucinations is critical
- You have retrieved context to verify against
- Trustworthiness is paramount

Accuracy

- You have ground truth answers
- Factual correctness is most important
- Evaluating knowledge-based systems

Hyperparameter Tuning

Hyperparameter	What it controls	Trade-offs
Chunk size & overlap	How big document pieces are, how much context each chunk has	Larger chunks → more context, fewer splits, but risk mixing irrelevant info; smaller chunks with overlap → better coverage but more redundancy / cost
Top-k (retrieved items)	How many retrieved chunks are fed to the generator	Too few → may miss needed info; too many → more noise and delay, maybe conflicting context

Practical Evaluation Steps

01

Prepare a test set

queries + reference answers + known relevant documents/chunks

02

Evaluate retriever separately

using Precision@k, Recall@k, MRR, MAP

03

Evaluate generation

compare generated answers to references using ROUGE, BLEU, BERTScore

04

Check faithfulness / hallucinations

see if statements are supported by retrieved context; possibly have human or LLM judge

05

Experiment

vary chunk size, overlap, top-k → observe how metrics change