Retrieval-Augmented Generation Done Right: What Actually Improves Accuracy?

A portfolio-style undergraduate research report and ablation blueprint

Author: Jaivik Joshi

Date: February 11, 2026

Abstract. Retrieval-Augmented Generation (RAG) improves factual answering by combining a retriever (non-parametric memory) with a generator (a language model). In practice, RAG quality depends on many engineering decisions: how documents are split into chunks, which embedding model powers dense retrieval, whether a reranker is used, and how the prompt instructs the generator to use retrieved evidence. This report presents (1) a controlled ablation framework that isolates the contribution of each component on a Wikipedia-backed QA benchmark, (2) a results table template and analysis checklist for interpreting outcomes, and (3) practical best practices for building reliable RAG systems under realistic constraints. Where numeric results are shown, they are labeled as *illustrative* (placeholders) unless explicitly reproduced on the stated dataset.

Reader note (portfolio transparency). This document is written to be publishable on a personal website. It is intentionally explicit about which parts are executed measurements versus planned experiments. If you replicate the matrix in Section 4, you can replace illustrative table values with your own measured numbers and keep the narrative structure unchanged.

1. Introduction

Large language models (LLMs) can produce fluent answers, but factual correctness can degrade when information is (a) missing from training data, (b) too recent, or (c) difficult to recall precisely. Retrieval-Augmented Generation (RAG) addresses this by fetching relevant passages from an external corpus (e.g., Wikipedia or product documentation) and conditioning generation on those passages. The core promise is straightforward: improve correctness and provide grounding by giving the model access to explicit evidence.

Research question. In a standard RAG pipeline, which component most strongly affects factual QA accuracy: (1) chunking strategy, (2) embedding model, (3) reranking, or (4) prompting?

Contributions. This report provides: (a) a controlled ablation methodology designed to be reproducible, (b) a compact but practical taxonomy of failure modes (retrieval miss, context dilution, hallucination, and instruction drift), and (c) best-practice recommendations prioritized by expected impact and cost.

2. Background

A RAG system typically has two phases. During indexing, documents are cleaned, split into chunks, embedded into vectors, and stored in a vector index. During answering, a query is embedded, top-k chunks are retrieved by similarity, optionally reranked with a more precise model, and inserted into a prompt for an LLM to generate an answer. The original RAG formulation (Lewis et al., 2020) frames retrieval as access to non-parametric memory paired with a sequence-to-sequence generator.

3. Related Work

RAG was formalized for knowledge-intensive NLP tasks by Lewis et al. (2020). Dense retrieval and sentence embedding techniques are commonly powered by bi-encoders such as Sentence-BERT (Reimers & Gurevych, 2019). In many applied systems, cross-encoder rerankers improve retrieval precision by jointly encoding query and candidate passages; BERT-based reranking is a widely cited approach (Nogueira & Cho, 2019). Finally, evaluating end-to-end RAG requires measuring both retrieval quality and generation faithfulness; RAGAS proposes reference-free evaluation dimensions such as faithfulness and context relevance (Es et al., 2024).

4. Experimental Design (Controlled Ablation)

This section describes a concrete experiment plan you can execute with Python and common open-source tooling. The intent is to hold all variables fixed except one factor at a time, so effect sizes are interpretable.

Dataset. Use a Wikipedia-grounded open-domain QA benchmark such as *Natural Questions (open)* or *HotpotQA*.

For a lightweight portfolio replication, sample 500-2,000 questions to keep evaluation manageable while still stable. Ensure the retrieval corpus is aligned with the dataset: either a Wikipedia dump consistent with the benchmark, or a curated subset produced from the dataset's provided contexts.

4.1 Evaluation Metrics

- Answer accuracy: exact match (EM) or token-level F1.
- Retrieval recall@k: whether a passage containing the gold answer is in top-k.
- MRR: mean reciprocal rank of the first answer-containing chunk.
- Faithfulness: whether the generated answer is supported by retrieved context (e.g., RAGAS).
- Cost/latency: optional but practical for engineering recommendations.

4.2 Controls

Keep constant: generator model, decoding settings (temperature and max tokens), k (top-k retrieved), context window budget, and the evaluation script. Change only one axis per ablation.

4.3 Ablation Matrix

Factor	Variants (examples)	What you hold constant	Primary hypothesis
Chunking	256 / 512 / 1024 tokens; 10-20% overlap; section/page splits	Embedder, retriever, reranker, prompt	Mid-size chunks balance recall and dilution
Embedding	MiniLM / MPNet / BGE-large; (optional) API embeddings	Chunking, reranker, prompt	Stronger embeddings improve recall@k most
Reranking	None vs cross-encoder rerank top-50 -> top-5	Chunking, embedder, prompt	Reranking boosts precision and final EM
Prompting	Naive vs grounded prompt with refusal policy; optional citations	Chunking, embedder, reranker	Better instructions reduce hallucinations

Table 1. Recommended ablation matrix for a reproducible RAG study.

4.4 Reproducible Defaults (Implementation)

A straightforward implementation can be built with: a document loader, a text splitter (token-based or structure-based), an embedding model (SentenceTransformers), a vector index (e.g., FAISS), an optional cross-encoder reranker, and a generator. Use deterministic generation settings (temperature=0 or close to 0) during evaluation.

Prompt baseline. Provide retrieved passages, then ask the question. **Prompt grounded.** Add: “Use only the provided context. If insufficient, respond ‘I don’t know.’” Optionally require short citations (e.g., [doc2]).

5. Results (Template + Illustrative Example)

The numbers in Table 2 are *illustrative placeholders* (not measured in this document). Replace them with your outputs.

Setting	EM/Acc	Recall@5	MRR	Faithfulness	Notes
Baseline: 512 tok, MiniLM, no rerank, naive prompt	0.73	0.80	0.62	0.84	Common starting point
Chunk 256 tok (overlap 15%)	0.69	0.82	0.58	0.83	Fragmentation; misses multi-sentence answers
Chunk 1024 tok (overlap 10%)	0.75	0.78	0.63	0.85	More context; some dilution
Embedding upgrade (e.g., MPNet/BGE)	0.81	0.86	0.71	0.86	Recall boost
Add rerank (top50->top5)	0.86	0.86	0.78	0.88	Precision + EM jump
Grounded prompt + refusal	0.78	0.80	0.62	0.91	Hallucinations drop; more abstains
Best combo (embed + rerank + grounded)	0.92	0.90	0.83	0.93	High accuracy and grounding

Table 2. Example results table (illustrative placeholders).

5.1 Interpreting Outcomes

When accuracy changes, determine whether gains came from retrieval (Recall@k / MRR) or generation (faithfulness). Embedding upgrades typically raise recall, while grounded prompting improves faithfulness without changing retrieval. Reranking often increases MRR, placing answer-bearing passages earlier and reducing context needs.

5.2 Statistical Reporting (Optional)

For portfolio credibility, include confidence intervals via bootstrap resampling over questions (e.g., 1,000 resamples). Report mean EM and 95% intervals to communicate uncertainty and avoid over-claiming small differences.

6. Error Analysis

Pair quantitative metrics with qualitative examples that map failures to causes. Sample ~50 errors, label a primary failure mode for each, and summarize proportions.

- **Retrieval miss:** answer-containing passage absent from top-k.
- **Fragmentation:** answer spans chunk boundaries.
- **Context dilution:** retrieved chunks contain strong distractors.
- **Instruction drift:** model ignores “use context only.”
- **Hallucination:** model invents details not present in context.

7. Best Practices (Actionable Checklist)

Highest-leverage sequence: (1) set chunking baseline, (2) test two embedding models, (3) add a reranker, (4) tighten prompts to enforce grounding, and (5) measure retrieval + faithfulness.

Priority	Recommendation	Why it helps
High	Add a cross-encoder reranker (top50->top5)	Improves precision; surfaces answer-bearing passages
High	Test stronger embedding models (MPNet/BGE/API)	Boosts recall; fewer retrieval misses
Medium	Use 500-1,000 token chunks with 10-20% overlap	Balances fragmentation vs dilution
Medium	Grounded system prompt + refusal policy	Reduces hallucination; improves trustworthiness
Low	Optimize prompt formatting and citation tags	Improves debugging and readability

Table 3. Practical best-practices checklist.

8. Limitations

RAG results are sensitive to dataset choice, corpus preprocessing, evaluation metrics, and the generator model. Ablations can shift under different document styles (short FAQs vs long manuals). For portfolio work, avoid claiming absolute SOTA numbers unless you can share code, seeds, and exact dataset splits.

9. Future Work

After the core ablation, extend the study with: (1) hybrid retrieval (BM25 + dense), (2) query rewriting, (3) multi-hop retrieval for compositional questions, and (4) context compression (summarize retrieved passages before prompting).

10. Conclusion

RAG accuracy is shaped by a chain of components: chunking determines retrieval units, embeddings determine recall, reranking determines precision, and prompting determines whether generation stays faithful to evidence. A controlled ablation study attributes gains to specific choices, producing a credible portfolio artifact.

Appendix A. Prompt Templates

Naive prompt: Context chunks + Question + “Answer using the context.”

Grounded prompt (recommended):
System: You are a factual assistant. Use only the provided context. If insufficient, say “I don’t know.”
User: Context: [doc1] ... [dock] Question: ...

Appendix B. Reproducibility Checklist

- Fix random seeds and record software versions.
- Record chunk size, overlap, tokenizer, and context token budget.
- Record embedding model name and index settings (metric, FAISS type).
- Record reranker model and candidate pool size (e.g., 50).
- Record prompt, decoding settings, and evaluation scripts.

Appendix C. Example Experiment Log Sheet

Keeping an experiment log makes your results auditable. Below is a compact schema you can copy into a spreadsheet or a JSONL file. Each row corresponds to one run in your ablation matrix.

run_id	chunk_size	overlap	embed_model	reranker	prompt_version	k	EM	Recall@5	notes
001	512	0.15	all-MiniLM-L6-v2	none	naive	5			baseline
002	512	0.15	bge-large-en	none	naive	5			embedding upgrade
003	512	0.15	bge-large-en	cross-encoder	grounded	5			full combo

Table C1. Example experiment log schema (leave metric cells blank until you run experiments).

References (APA 7)

Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAS: Automated evaluation of retrieval augmented generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. ACL Anthology.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems (NeurIPS).

Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).