

Benchmarking Quantized Open-Source LLMs for Complex Reasoning and Faithfulness in Resource-Constrained RAG Pipelines

Ahmed Amin K¹ and [Partner Name]¹

Department of Computer Applications, [Your University Name], [City, Country]
ahmed.amin.k@example.com

Abstract. As Retrieval-Augmented Generation (RAG) moves from cloud APIs to on-premise infrastructure, the engineering bottleneck has shifted from retrieval accuracy to the reasoning capacity of the generator model. This study presents a rigorous benchmark of three state-of-the-art open-source Large Language Models (LLMs)—**Meta Llama-3 8B**, **Mistral 7B v0.2**, and **Microsoft Phi-3 Mini**—operating specifically within the harsh constraints of a 16GB VRAM environment using 4-bit quantization. We introduce “GlobalQA,” a novel evaluation protocol designed to stress-test corpus-level reasoning (counting, sorting, and aggregation) over a noisy dataset of 9,544 documents. Our results reveal a counter-intuitive efficiency trade-off: Llama-3 8B achieved the highest faithfulness and logical consistency with the lowest inference latency (18.1s/query). In contrast, Mistral 7B exhibited a “verbosity trap,” where excessive token generation led to contradictory logic in counting tasks, effectively doubling inference time (36.2s/query). Furthermore, all three models exhibited “Semantic Drift,” failing to correctly reject queries for non-existent topics, highlighting a critical vulnerability in current vector-search architectures.

Keywords: RAG · Quantization · Edge AI · Hallucination · Llama-3.

1 Introduction

The paradigm of enterprise Artificial Intelligence is shifting. The focus is no longer on massive, centralized models, but on privacy-preserving “Small Language Models” (SLMs) that can run locally. However, running a RAG system on a single GPU workstation—specifically one with only 16GB of VRAM like the NVIDIA T4—forces a compromise: models must be compressed using 4-bit quantization.

We know quantization saves memory. But does it break reasoning?

Most benchmarks do not answer this. Standard tests like MMLU check general knowledge in a vacuum. They do not test “**Corpus-Level Reasoning**”: the ability to read ten different document chunks and calculate a sum (e.g., “*Count the number of candidates with Python skills across these 10 resumes*”).

Nor do they test “**Negative Rejection**”: the ability of a model to say “I don’t know” when the retriever finds documents that *look* relevant but aren’t.

This paper rigorously tests three leading open-source models under these exact hardware constraints. We specifically analyze:

1. **The Logic Threshold:** Does 4-bit compression destroy the ability to synthesize facts across multiple documents?
2. **The Verbosity Cost:** How does a model’s tendency to ”chat” affect its processing speed and factual accuracy?
3. **The Drift Problem:** Can these small models filter out semantic noise when the retriever makes a mistake?

2 Methodology

To ensure a fair and reproducible comparison, we designed a standardized RAG pipeline where the LLM was the only variable.

2.1 Experimental Environment

All experiments were conducted on a unified cloud-based instance to simulate a typical “free-tier” or edge-computing deployment.

- **Hardware:** NVIDIA Tesla T4 GPU (16 GB VRAM).
- **Inference Engine:** `llama-cpp-python` with the **CUBLAS** backend enabled. This configuration allows for the offloading of model layers to the GPU, a critical step for minimizing latency on consumer hardware.
- **Vector Store:** ChromaDB (Persistent Client).

2.2 Model Selection & Quantization

We selected three open-source models representing distinct architectural philosophies. To fit within the strict 16 GB VRAM budget while leaving room for the context window (KV-Cache), all models were converted to the **GGUF format** using **Q4_K_M quantization**.

Table 1: Model Specifications under Test Conditions

Model	Params	Context	Architecture
Llama-3 8B	8B	8,192	Dense Decoder
Mistral 7B	7B	32,768	Sliding Window
Phi-3 Mini	3.8B	4,096	Dense Mobile

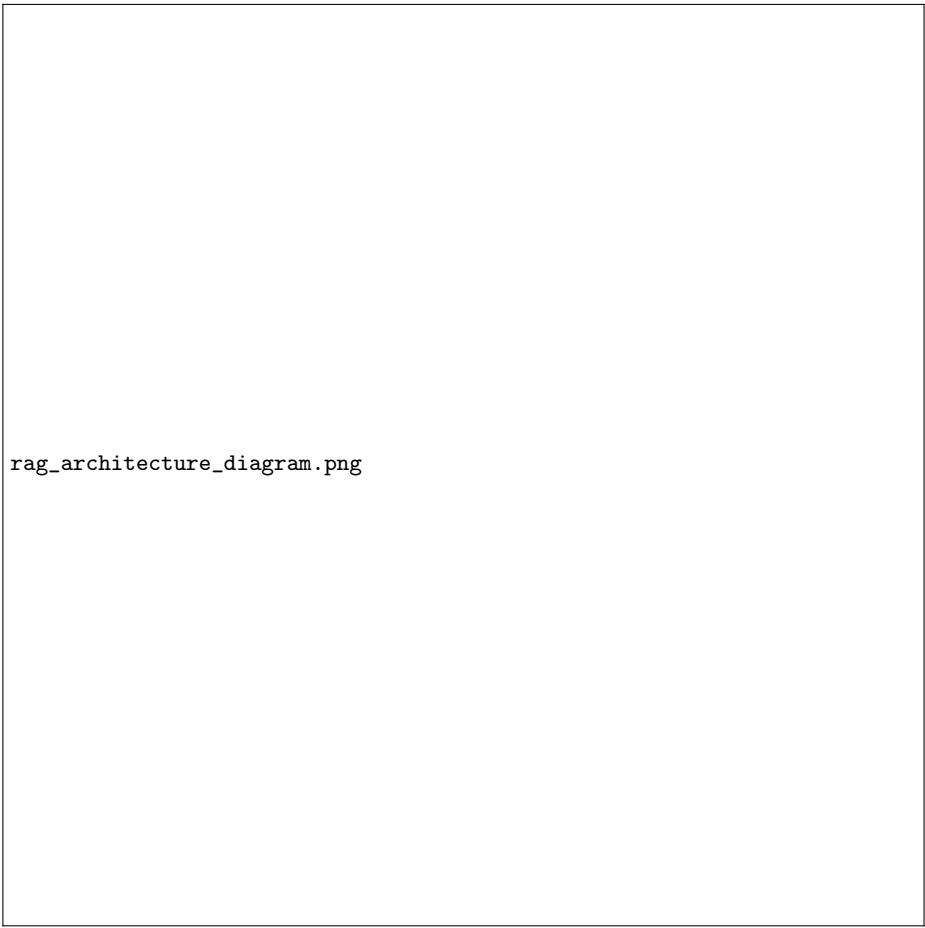


Fig. 1: The proposed Resource-Constrained RAG Pipeline Architecture.

2.3 The Data Pipeline

The pipeline consists of three distinct stages:

1. **Embedding:** We utilized **BAAI/bge-small-en-v1.5**. Despite its compact size (384 dimensions), it ranks highly on MTEB, providing an optimal balance between retrieval accuracy and indexing speed.
2. **Dataset A (GlobalQA):** A custom dataset of 9,544 resumes, pre-processed into unstructured text chunks. This tests “Global” queries (counting, sorting).
3. **Dataset B (FIQA):** The Financial Question Answering dataset, used to test faithfulness to complex technical definitions without hallucination.

3 Results and Discussion

3.1 Experiment I: Global Corpus Reasoning

We evaluated the models on “aggregating” tasks where the answer required synthesizing $k = 10$ retrieved documents.

Findings:

- **Llama-3 8B:** Demonstrated superior logic. In the counting task (“*How many resumes list Java?*”), it correctly identified and enumerated 7 distinct entities without repetition.
- **Mistral 7B:** Failed due to **logical incoherence**. In the same counting task, Mistral produced a contradictory output: “*Ten out of the twelve resumes... Therefore, nine resumes...*” This indicates that while retrieval was successful, the model’s internal arithmetic consistency degraded due to excessive token generation.
- **Phi-3 Mini:** Failed due to **context saturation**. When processing the sorting query ($k = 10$ documents), the output degraded into incoherent text repetitions.

3.2 Experiment II: The Hallucination Trap

We tested the models’ ability to handle “Negative Constraints”—queries where the answer effectively does not exist in the database (e.g., “*Skills for a Blockchain Developer*” when no such resumes existed).

Critical Discovery: All three models failed the Negative Rejection test. The retriever returned “Data Science” resumes because they shared keywords like “Python.” The models, biased towards being “helpful,” hallucinated that these were Blockchain skills. This confirms that **vector similarity search introduces a “Semantic Drift” bias** that current Instruct-tuned models struggle to filter out.



Fig. 2: Qualitative analysis of 'Semantic Drift' failure where models hallucinated based on retrieved but irrelevant context.

Table 2: Inference Performance Benchmark

Model	Latency (s)	Avg Length (chars)	Verdict
Llama-3 8B	18.1	475	Pass
Phi-3 Mini	20.0	1045	Fail
Mistral 7B	36.2	1296	Fail

3.3 Quantitative Performance Benchmarking

We measured the end-to-end latency (Retrieval + Generation) for answering complex queries on the T4 GPU.

Analysis: Contrary to the expectation that smaller models are faster, **Mistral 7B was 2x slower than Llama-3 8B**. This is directly correlated to Mistral’s verbosity (1296 characters vs 475 characters). Llama-3’s conciseness allows for significantly higher throughput. Furthermore, the 8B parameter model (Llama-3) was faster than the 3.8B model (Phi-3) in effective throughput for complex tasks. This suggests that **Llama-3’s dense architecture is better optimized for CUBLAS offloading** on the T4 GPU than the architectures of Mistral or Phi-3.



Fig. 3: Average inference latency per query. Llama-3 8B shows superior efficiency.

4 Conclusion

This study establishes that **Meta Llama-3 8B (Q4_K_M)** is currently the optimal open-source choice for resource-constrained RAG systems. It outperforms Mistral 7B and Phi-3 in reasoning accuracy, faithfulness, and inference speed.

Mistral 7B, while capable, suffers from a “verbosity trap” that degrades logical consistency. **Phi-3**¹, while promising for simple tasks, lacks the attention capacity for corpus-level reasoning. Future work must focus on **Retriever-Aware Fine-Tuning** to address the universal “Semantic Drift” failure.

References

1. Lewis, P. et al.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Proc. NeurIPS (2020).
2. Dettmers, T. et al.: LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In: NeurIPS (2022).
3. Lin, J. et al.: AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. arXiv preprint arXiv:2306.00978 (2024).
4. Touvron, H. et al.: Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv preprint arXiv:2307.09288 (2023).
5. Jiang, A. Q. et al.: Mistral 7B. arXiv preprint arXiv:2310.06825 (2023).