Multi-Model Question Answering Evaluation and Intelligent Routing

1. Introduction

Building upon the containerized Question Answering (QA) pipeline developed in Milestone 2, Milestone 3 advances the system toward large-scale evaluation, dataset-aware routing, and intelligent model selection. While Milestone 2 focused on establishing a modular docker container-based architecture and baseline evaluation, Milestone 3 emphasizes data-centric optimization and routing intelligence across diverse QA datasets.

The objective of this milestone is to evaluate how dataset characteristics, context length, and prompt complexity influence latency, accuracy, and overall utility, and to explore routing policies that dynamically select the most suitable model for a given query.

2. System Evolution

2.1 System Evolution from Milestone 2

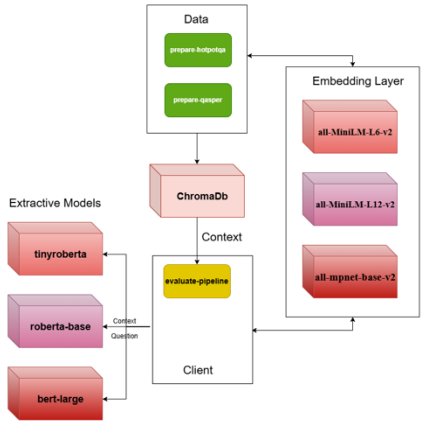

Figure 1 : Milestone 1 pipeline

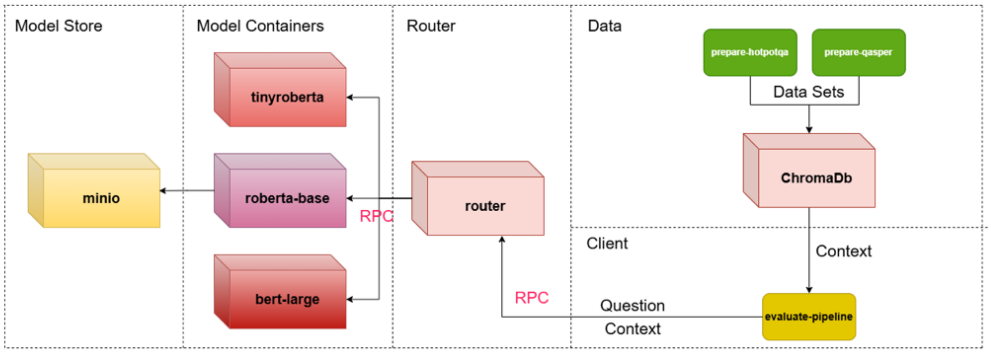

Figure 2 : Milestone 2 pipeline with Dockerization

In Milestone 2, we implemented:

- A fully containerized QA pipeline orchestrated via Docker Compose.
- Multiple transformer-based Extractive QA model variants (TinyRoBERTa, RoBERTa-Base, BERT-Large).
- A gRPC-based router service for dispatching requests to model backends.
- Dataset preparation pipelines for HotpotQA and QASPER.
- A shared ChromaDB instance for embedding storage and retrieval.
- An evaluation container capable of measuring accuracy, latency, and throughput.

This milestone established a **modular and extensible baseline** upon which more advanced routing and evaluation strategies could be developed.

2.2 Results from Milestone 2

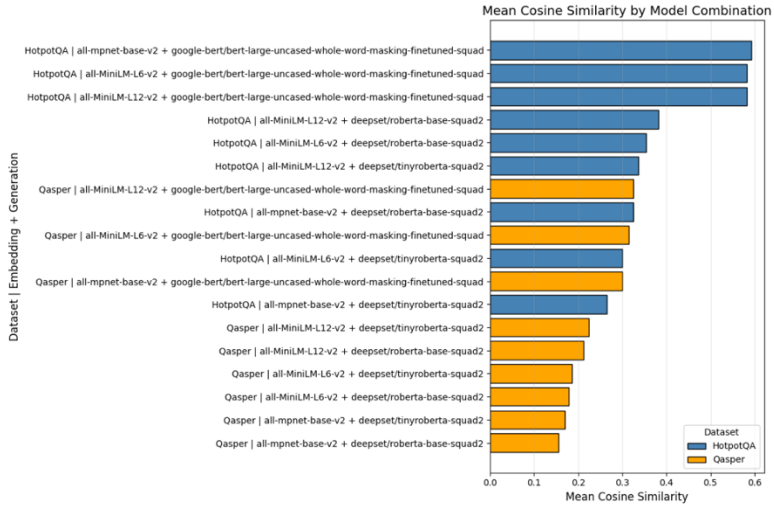

Figure 3 : Milestone 2 results

Figure 3 from Milestone 2, compares mean cosine similarity (answer quality) across different embedding–generation model combinations for the HotpotQA and QASPER datasets. Overall, HotpotQA consistently achieves higher cosine similarity scores than QASPER. The best-performing configurations generally pair stronger embedding models (e.g., all-mpnet-base-v2 or all-MiniLM-L12-v2) with extractive models such as BERT-large, fine-tuned highlighting the importance of both high-quality retrieval representations and powerful models.

In contrast, QASPER results show lower and more compressed scores across combinations. Replacing BERT-large with lighter generators (e.g., TinyRoBERTa, RoBERTa variants) leads to a drop in similarity.

3. Milestone 3 Architecture Overview

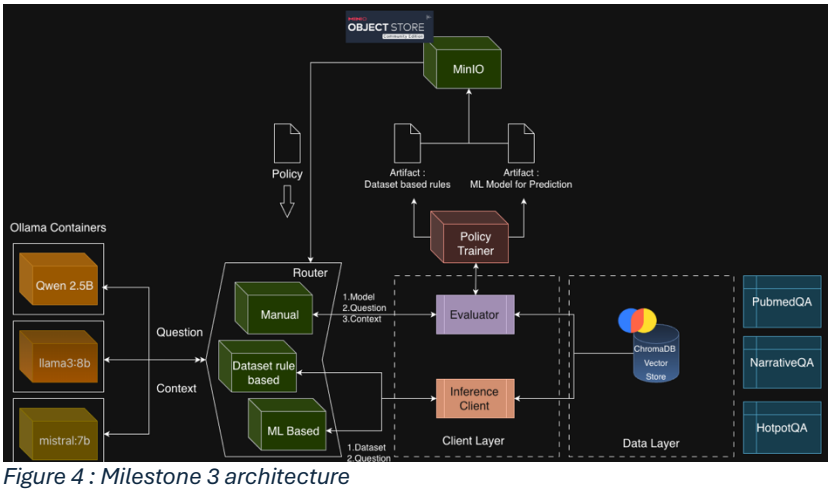

Figure 4 : Milestone 3 architecture

Milestone 3 introduces a second-generation pipeline that improves scalability, dataset diversity, and routing intelligence. The data layer was expanded to support multiple heterogeneous QA datasets, including, HotpotQA, NarrativeQA and PubMedQA. A total of 10,000 samples per dataset were ingested, yielding 30,000 rows stored using a unified schema and indexed with ChromaDB vector search. This enables consistent retrieval behavior across datasets.

Instead of extractive models , we have switched to small language modes. We have used Qwen2.5B, Lllama3-8B and Mistral-7B parameter models. We use Ollama as a framework to run these models in Dockerized containers.

We divided our experiments to two stages.

1. Stage 1: Training phase

We used 5,000 samples per datasets for evaluation. Every question was answered by all available models, enabling direct comparison. Latency was measured both with and without retrieval. Answer quality was evaluated using cosine similarity against gold-standard answers. This setup provided fine-grained insights into performance trade-offs across datasets, models, and context sizes

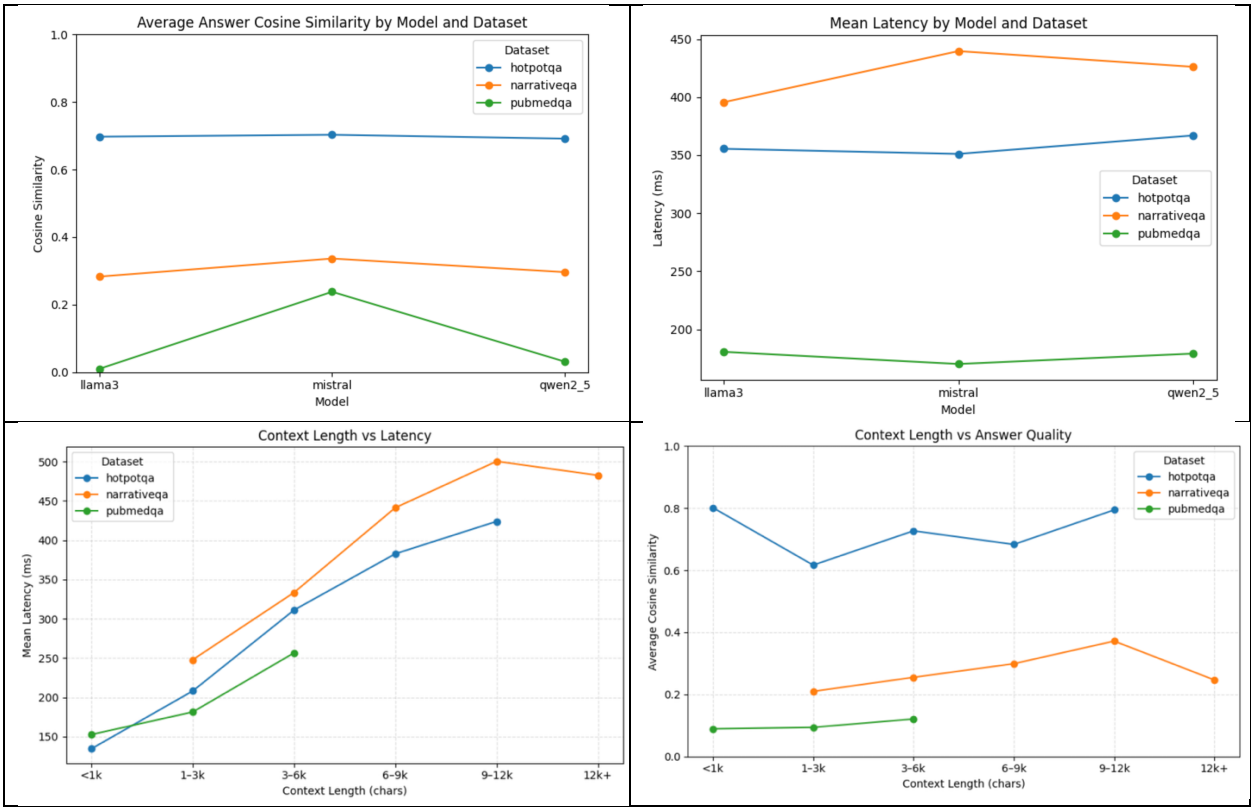

Figure 5 : Training stage results

The evaluation revealed that model performance is highly dataset dependent. Qwen and Llama models performed consistently well on factoid-style datasets. Mistral were better suited for long-form narrative reasoning tasks. No single model significantly dominated across all datasets.

These findings motivated the exploration of dataset-aware routing policies and machine learning based routing policies.

2. Stage 2: Learning the routing policy and inference time routing

```
rows: 45000
unique questions: 15000
datasets: 3
models: ['llama3', 'mistral', 'qwen2_5']

Default (global best by mean EM): qwen2_5

=== Best model per dataset ===
    dataset best_model  mean_exact_match  mean_cosine
   hotpotqa    qwen2_5             0.464     0.691563
narrativeqa    mistral             0.000     0.336478
   pubmedqa    mistral             0.000     0.238067

=== Full leaderboard (top 3 per dataset) ===
    dataset   model  mean_exact_match  mean_cosine
   hotpotqa qwen2_5            0.4640     0.691563
   hotpotqa  llama3            0.4504     0.697455
   hotpotqa mistral            0.4348     0.703102
narrativeqa  llama3            0.0000     0.283110
narrativeqa mistral            0.0000     0.336478
narrativeqa qwen2_5            0.0000     0.296356
   pubmedqa  llama3            0.0000     0.010169
   pubmedqa mistral            0.0000     0.238067
   pubmedqa qwen2_5            0.0000     0.031367

Example routing:
  hotpotqa        -> qwen2_5
  narrativeqa     -> mistral
  pubmedqa        -> mistral
```

Figure 6 : Learned policy for dataset based routing

A dataset-based routing policy was implemented where; Incoming questions are classified by dataset type. We utilized cosine-similarity, and exact match to find best serving models for each given datasets as shown on Figure 6.

Based on the model learderboard (Figure 6) hotpotqa was best served by Qwen2 while NarrativeQA and PubmedQA were best served by mistral model. Initial results from the Dataset based routed had comparable results to the previous training stage.

Machine Learning–Based routing strategy was also explored using, features such as dataset type, prompt character length, context character length. Contrast to the dataset based policy where we only considered the dataset here we consider the dataset + properties of the question.A target variable (Best performing model for each question) was created using. A utility score indicating the best-performing model per question.

This approach achieved comparable results (Figure 7) to dataset-based routing for HotpotQA and PubMedQA, suggesting that feature engineering remains a critical improvement for effective ML-driven routing.

4.Discussion

Overall, several key findings emerge from this milestone. First, dataset characteristics significantly influence optimal model choice**,** confirming that a single model cannot universally perform best across all QA tasks. Second, context length introduces a clear accuracy–latency trade-off**,** where longer contexts improve answer quality but increase inference time. Third, data-centric routing strategies show strong potential, particularly for datasets with distinctive structures, though they require further refinement to generalize well. Fourth, while ML-based routing demonstrates promise, it currently needs richer semantic, retrieval, and workload-aware features.

5.Conlusion

Future work will focus on expanding the system to support eight unified QA datasets using the DoxplainQA schema available on HuggingFace. The backend will be migrated to AWS-hosted infrastructure to enable scalable experiments and SLA-driven optimization. Planned improvements include automated drift detection and model selection, the development of a third-generation frontend using React and Tailwind CSS, and enhancements to ML-based routing through richer semantic, retrieval-aware, and cost-sensitive features. In addition, cost-aware SLA compliance experiments will be conducted in cloud environments to better understand real-world deployment trade-offs.