Doxplain

Multi-Model QA Evaluation Dashboard

Code

https://github.com/deshanshehntha/Milestone-2

Team

Deshan
Microservices & HPC

Nagarjuna
Product Owner

Katelyn
Data Engineering & UI/UX

Where We Left Off...

Data Layer

HotpotQA & Qasper

Original schemas w/ custom data loaders

Direct loading from .parquet

Backend

1st gen pipeline

Test data, models, & encoders

Containerized with Docker

Frontend

1st gen UI

Display & compare datasets & models

Developed with Streamlit

Where We Are Now

Data Layer

HotpotQA & NarrativeQA, PubmedQA

Vector search enabled with ChromaDB

Backend

2nd gen pipeline

Fewer encoders, better models

Diverse data, faster ingestion

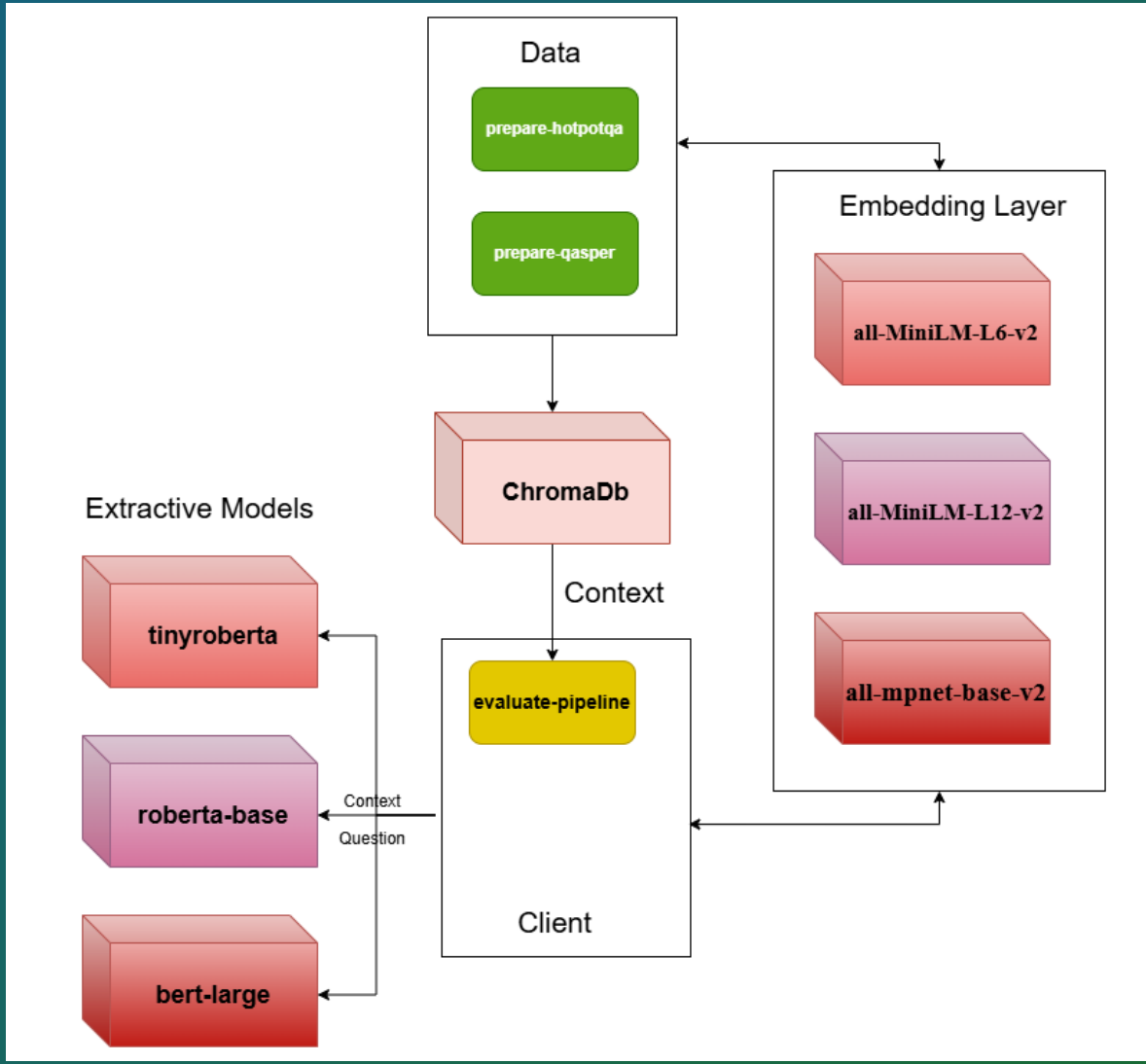
Frontend

2nd gen UI

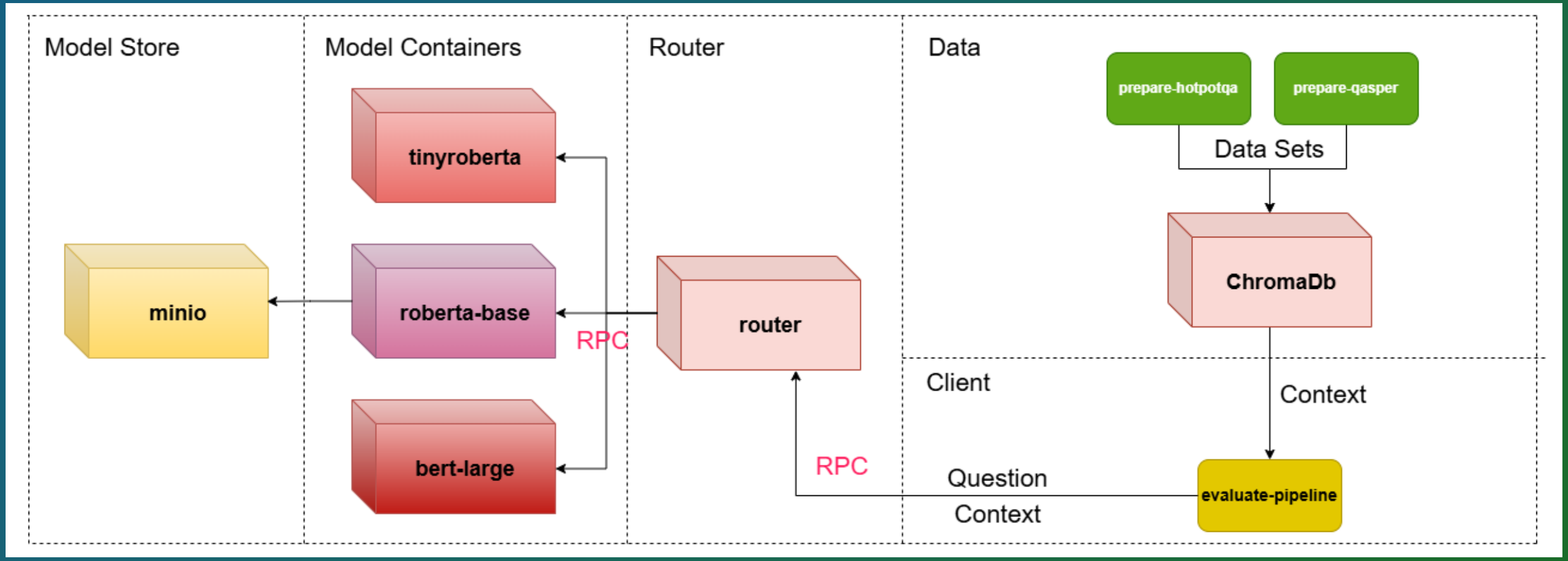
Multi-dataset experiment support

Live routing demo available

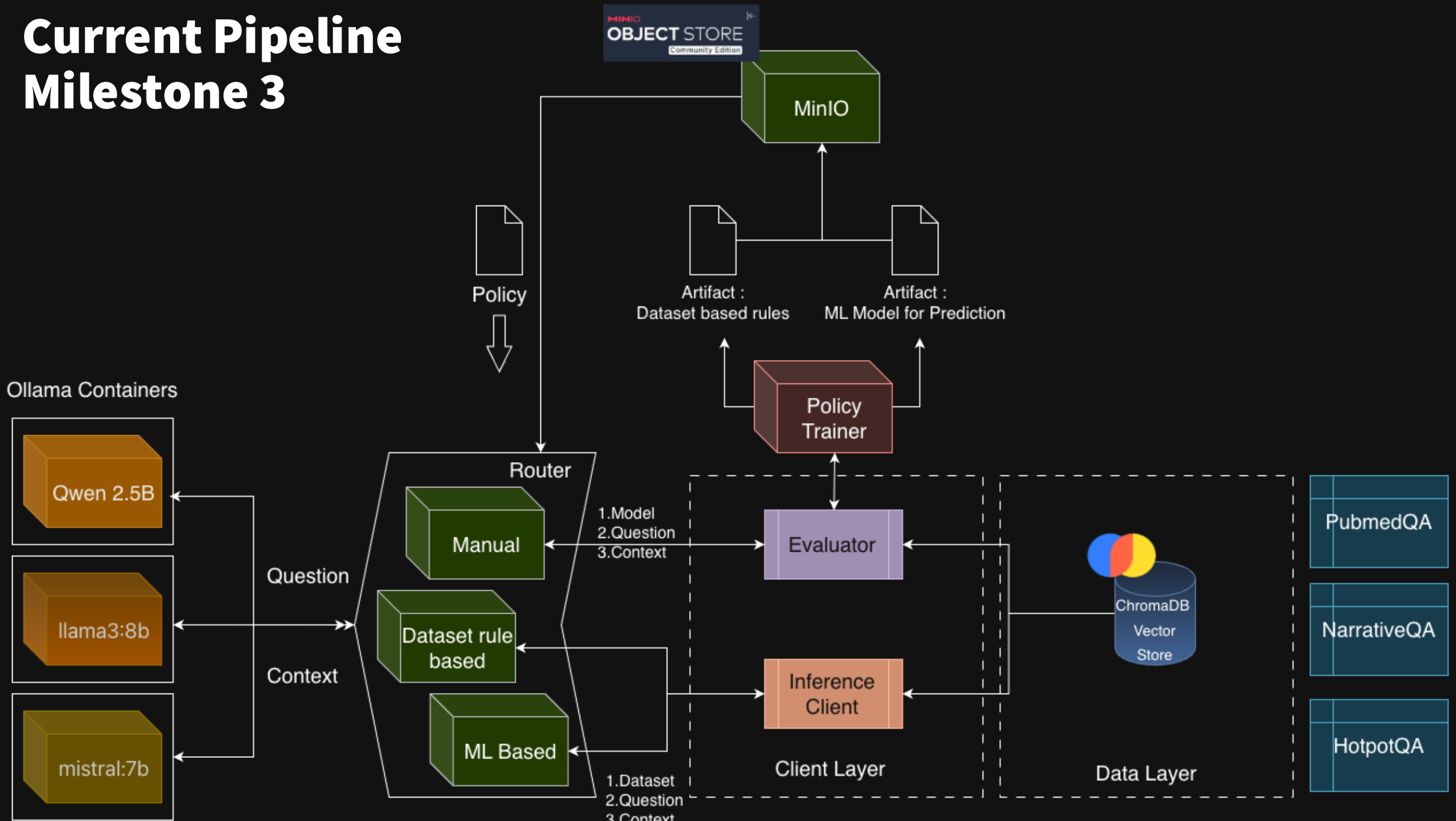
Pipeline Evolution
Milestone 1: Architecture



Pipeline Evolution
Milestone 2: Architecture



Current Pipeline Milestone 3



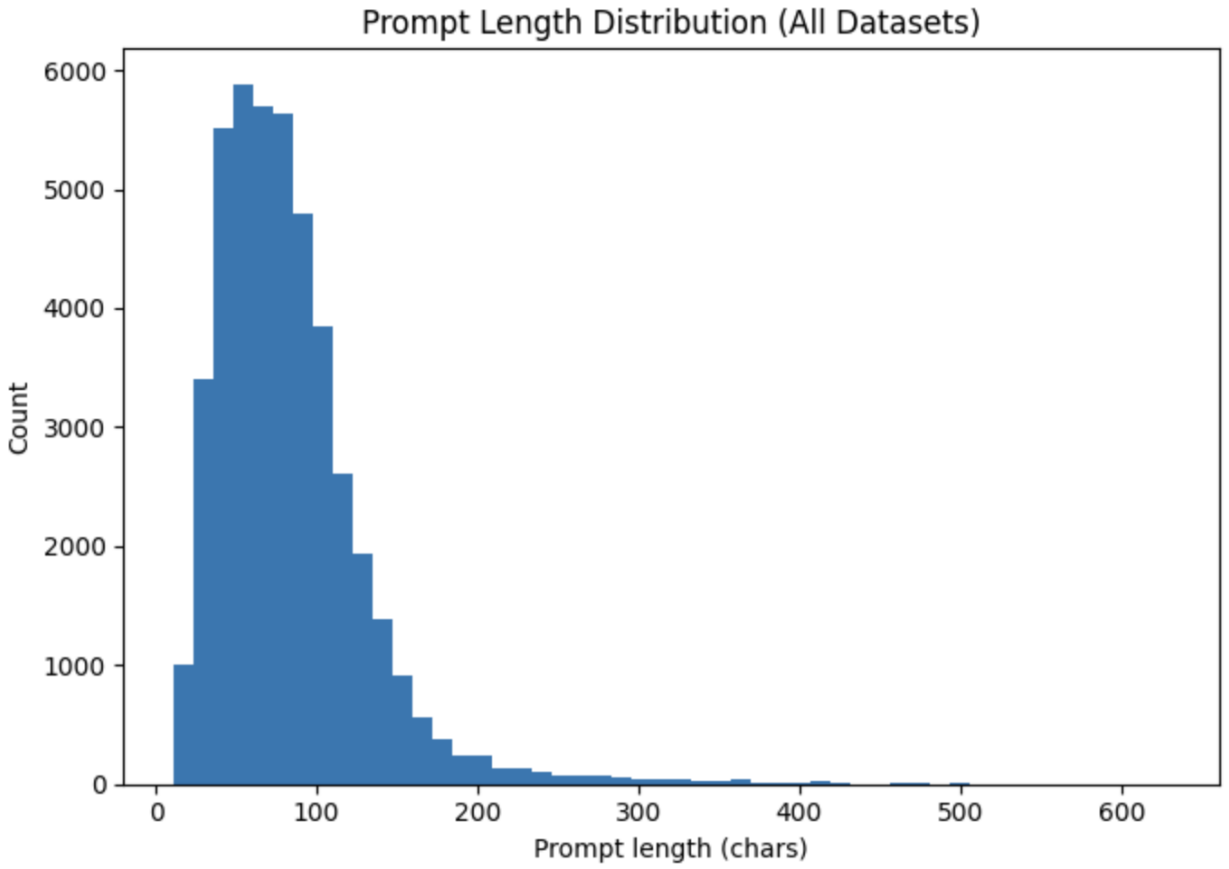
Data Layer

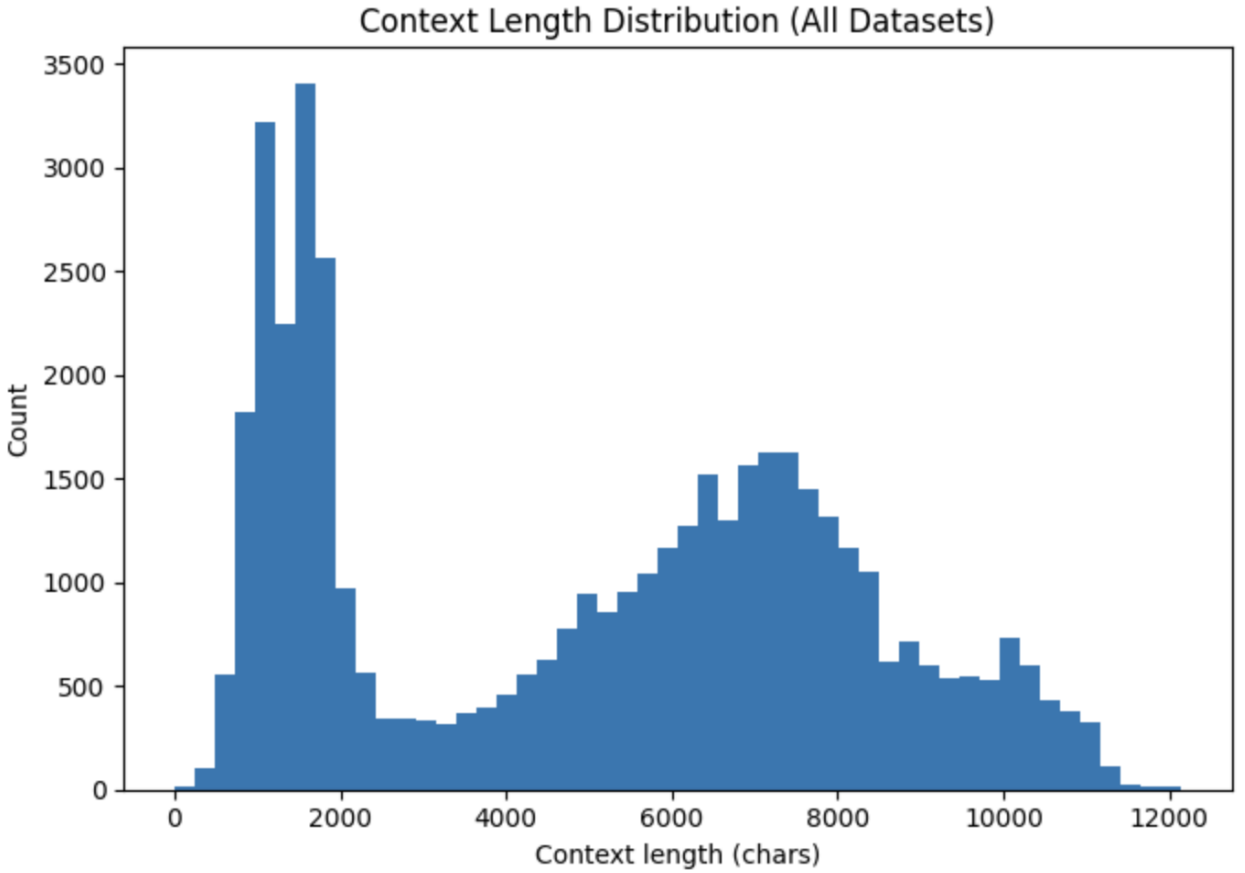
- 10000 rows from each dataset
  HotpotQA & NarrativeQA, PubmedQA

– 30,000 total rows

ChromaDB

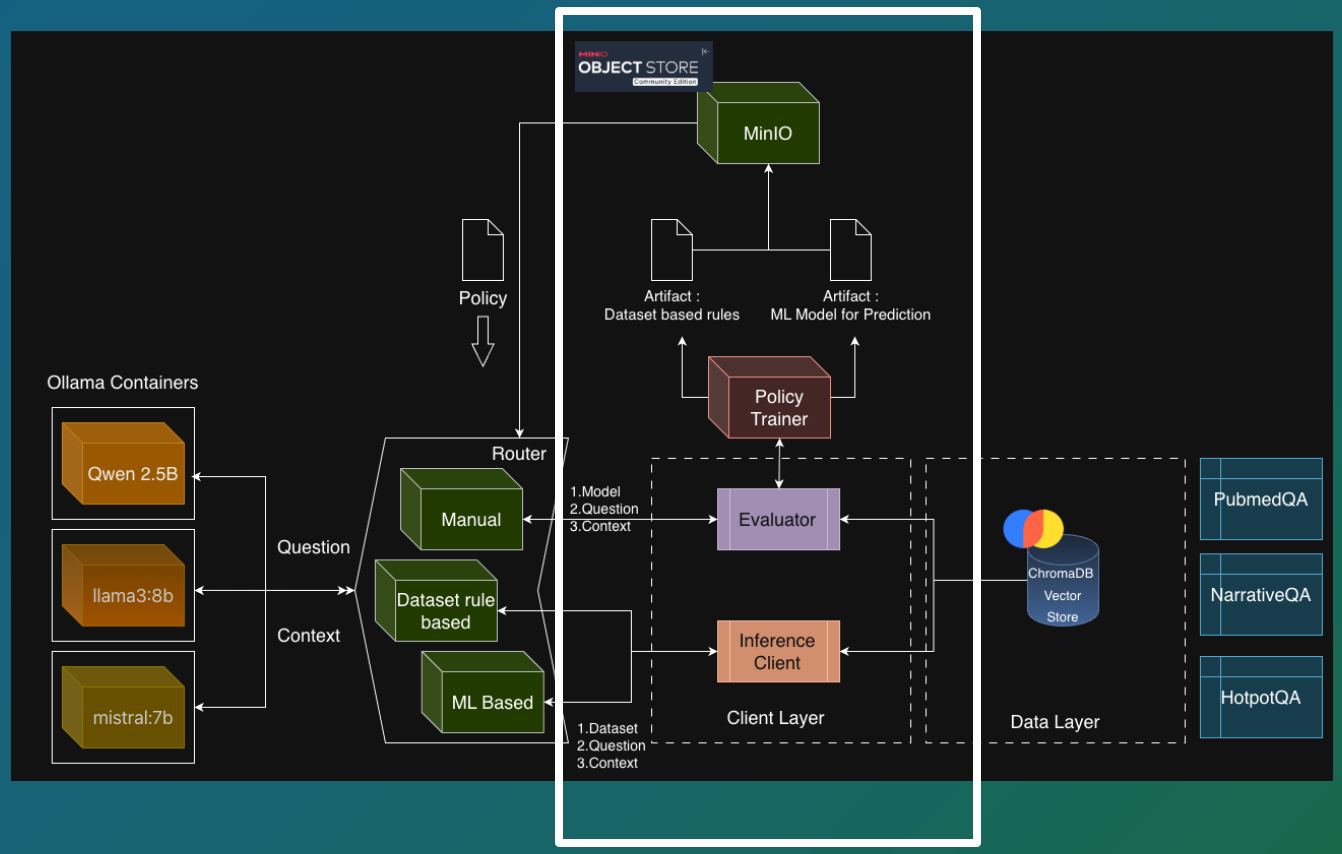
	HotpotQA	NarrativeQA	PubmedQA
No of Context Chunks	8836	4,438,678	72,373
CHAR_CHUNK_SIZE	2000	2000	2000
CHAR_CHUNK_OVERLAP	200	200	200
TOKEN_CHUNK_SIZE	256	256	256
TOKEN_CHUNK_OVERLAP	32	32	32
EMBED_MODEL	all-MiniLM-L6-v2	all-MiniLM-L6-v2	all-MiniLM-L6-v2





dataset	prompt_chars					context_chars				
	count	mean	median	min	max	count	mean	median	min	max
hotpotqa	15000	106.53	90.0	25	630	15000	6452.43	6604.5	413	9134
narrativeqa		48.08	46.0	11	160		7287.43	7821.0	0	12153
pubmedqa		95.86	93.0	11	329		1430.43	1451.5	236	4185

Client Layer – Evaluation Container



Client Layer

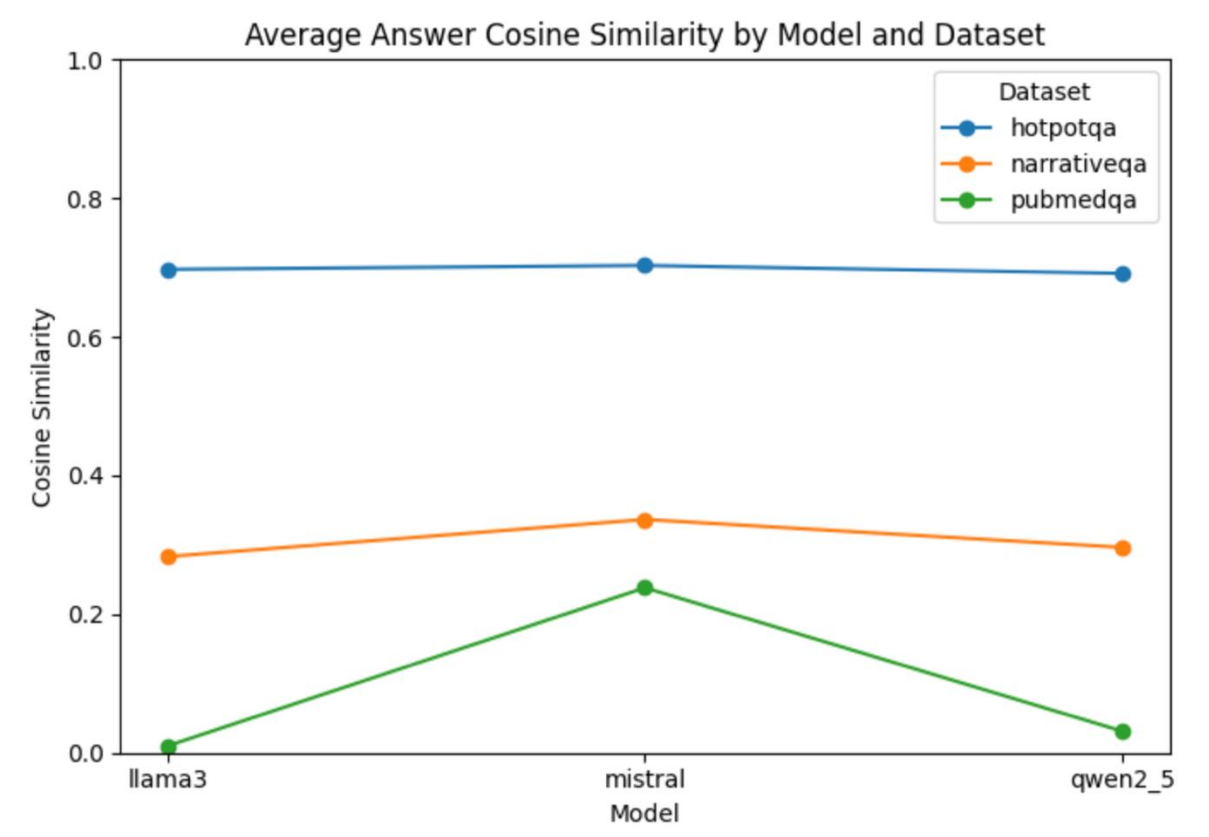
- 5000 rows from each dataset
  - HotpotQA
  - NarrativeQA,
  - PubmedQA

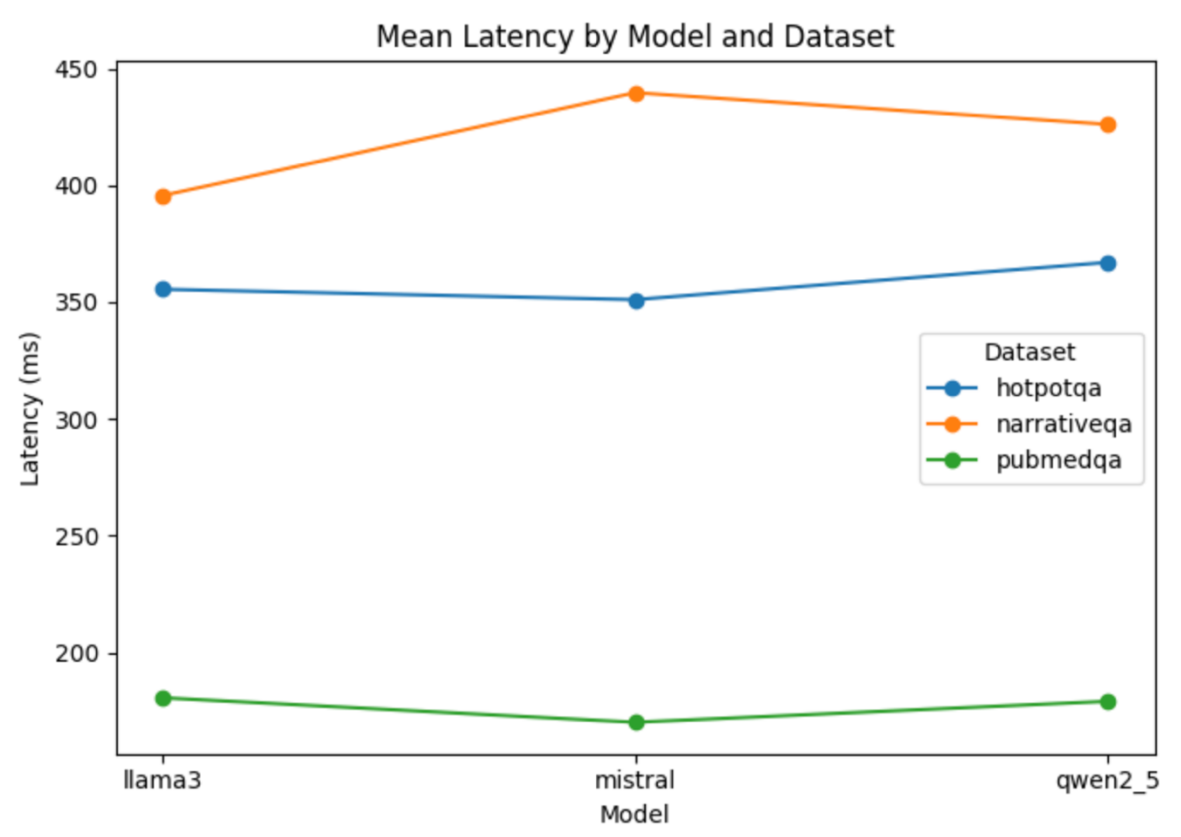
- 15,000 total rows

- All questions answered by all models

Client Layer – Evaluation container

Initial Results



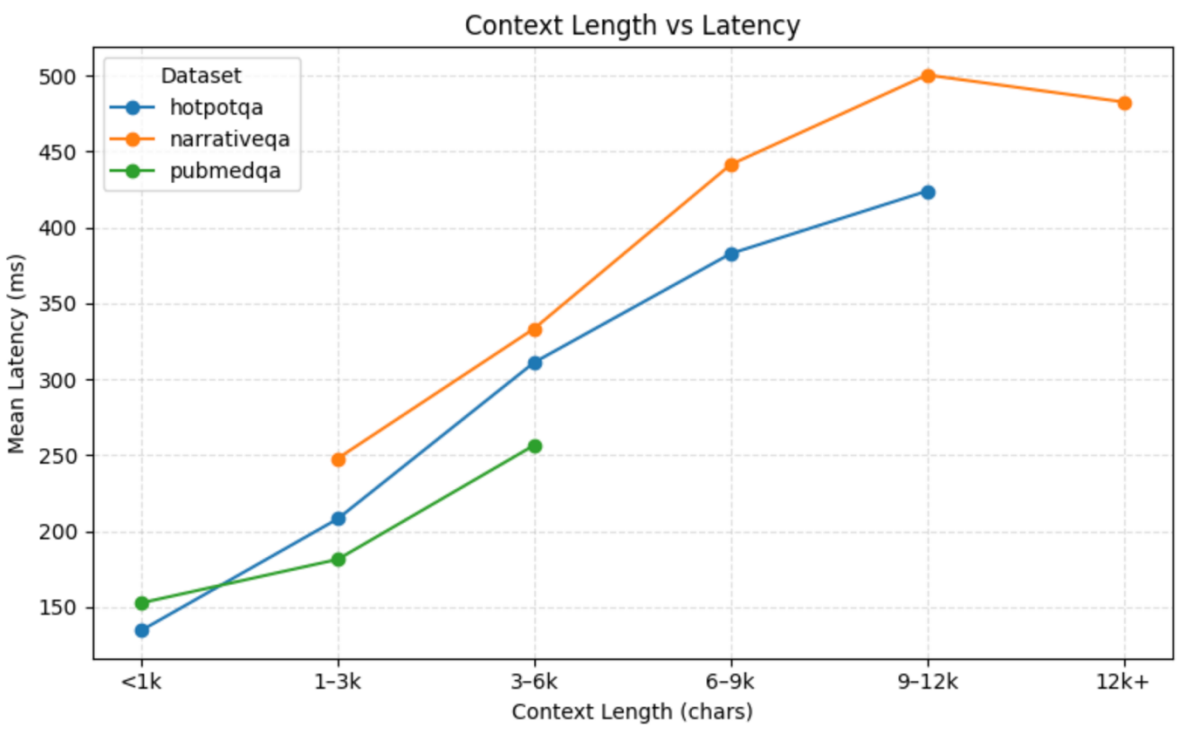


Cosine similarity Evaluation against gold standard answer

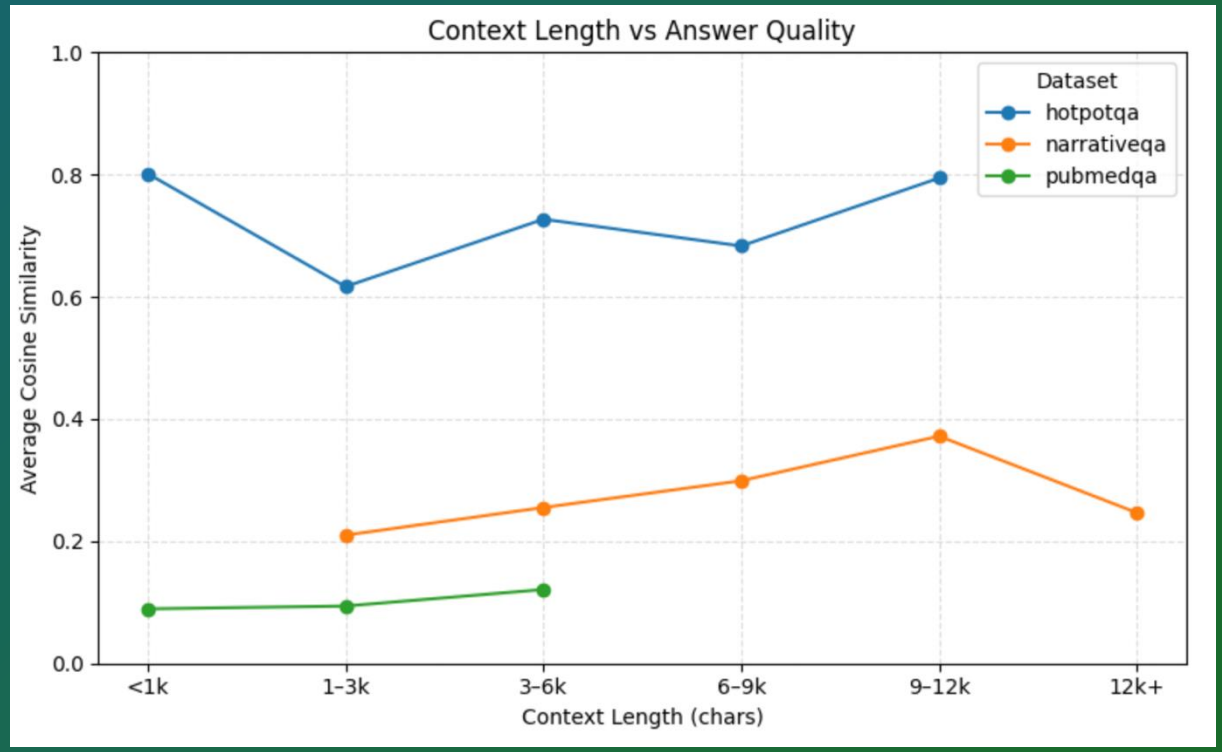
***Latency W/O retrieval**

Client Layer – Evaluation container

Initial Results



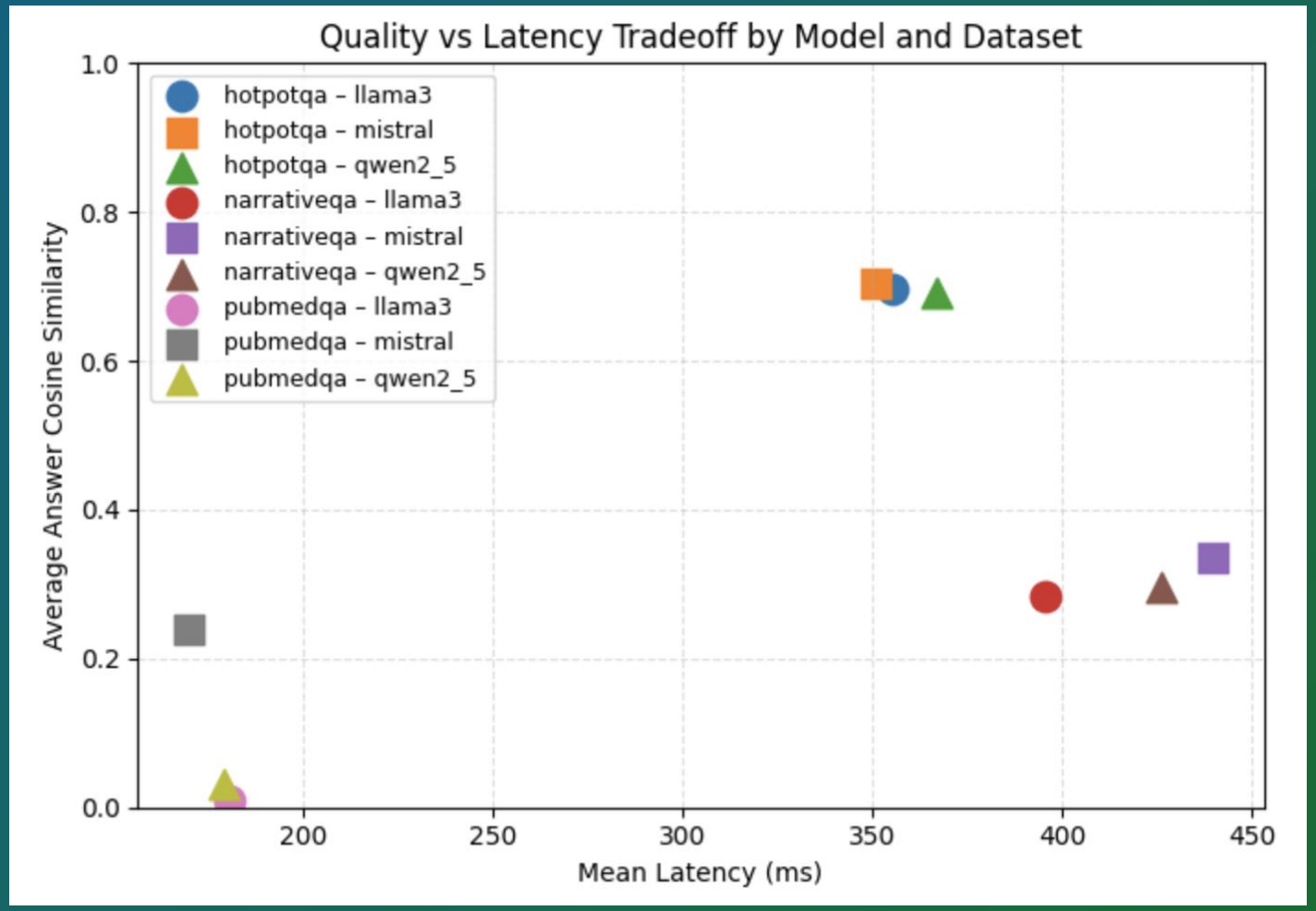
Context Length vs Latency



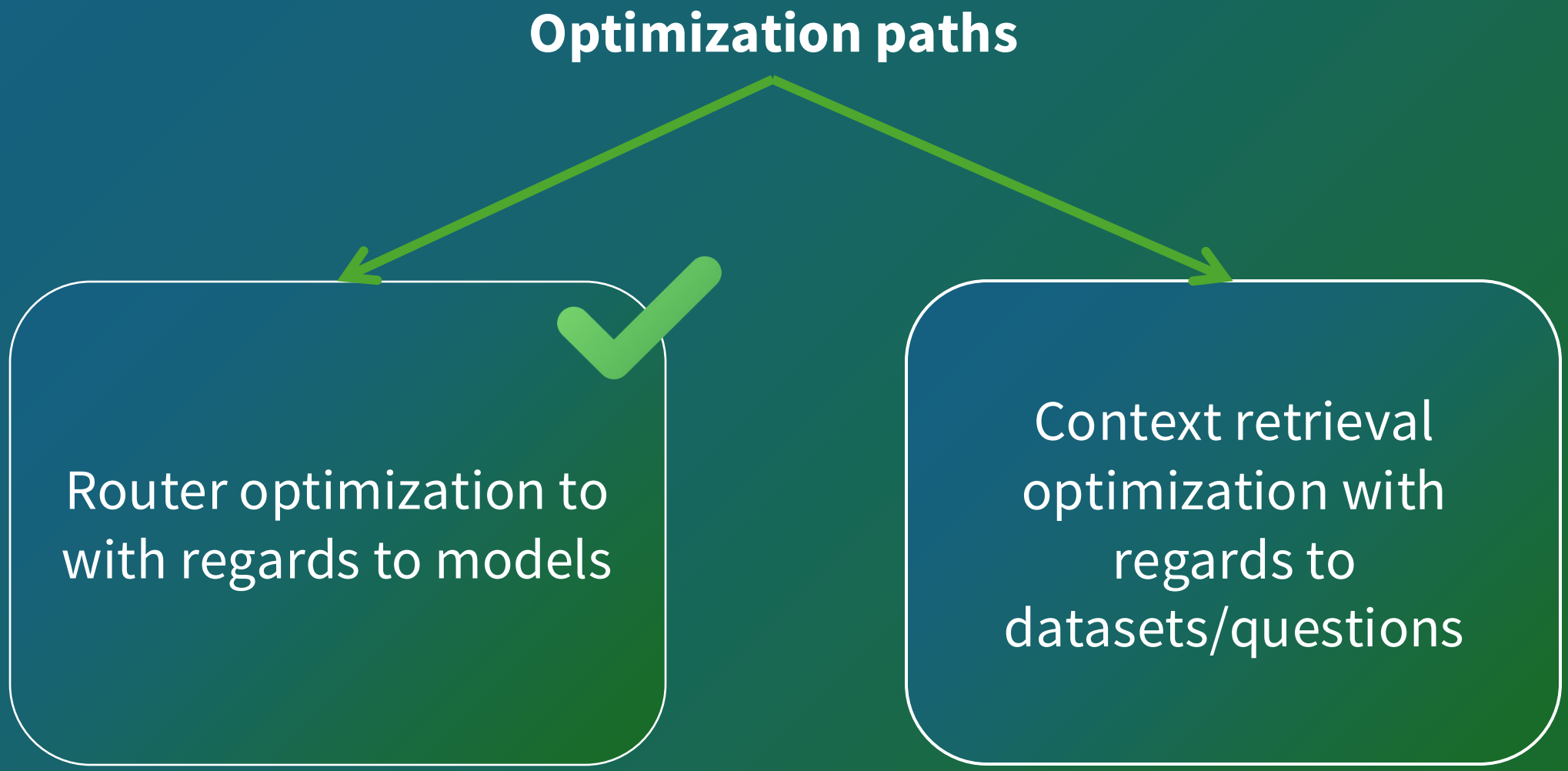
Context Length vs Answer Quality

Client Layer – Evaluation container

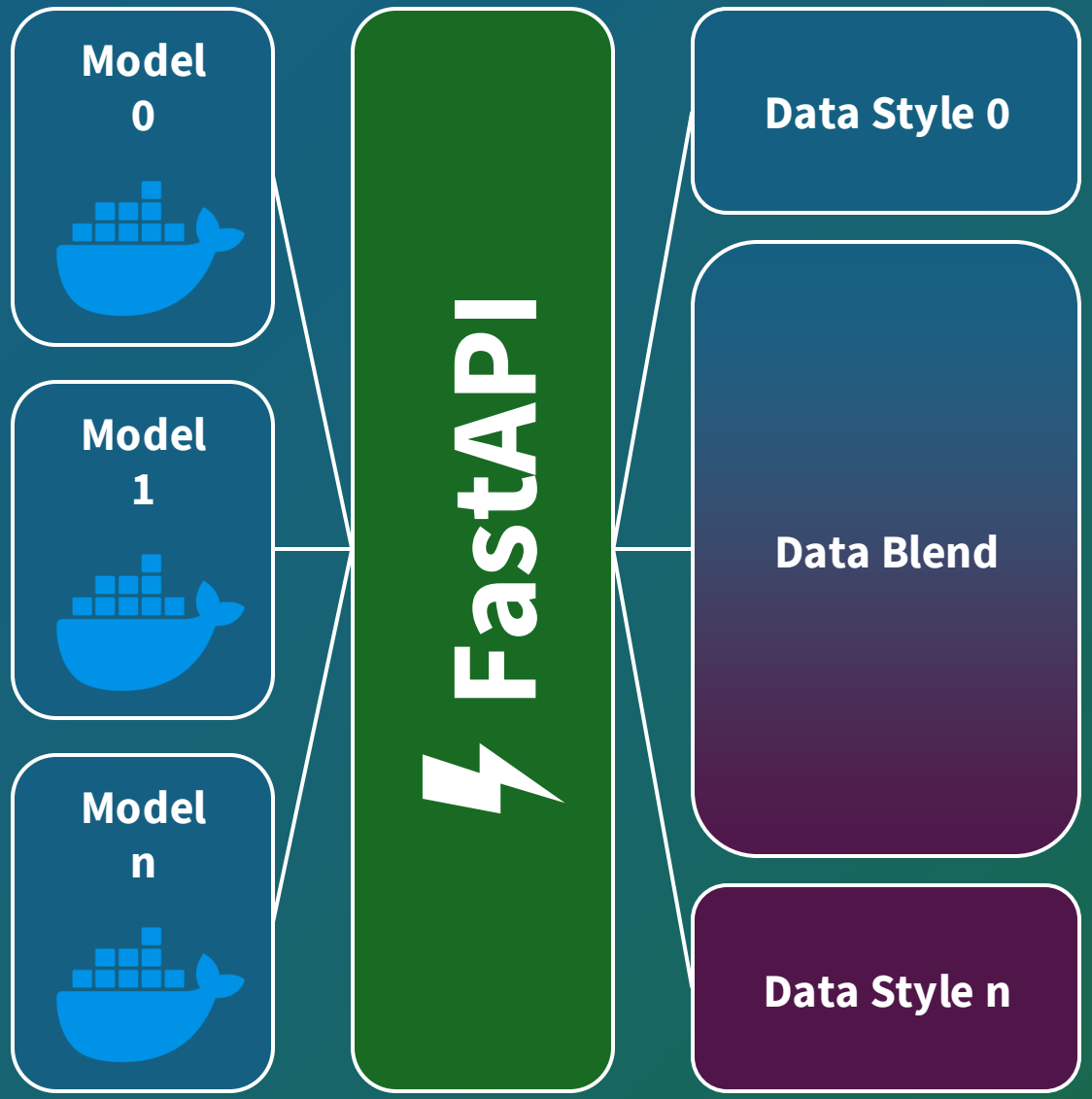
Initial Results



Based on Evaluation Results



Dataset based routing policy

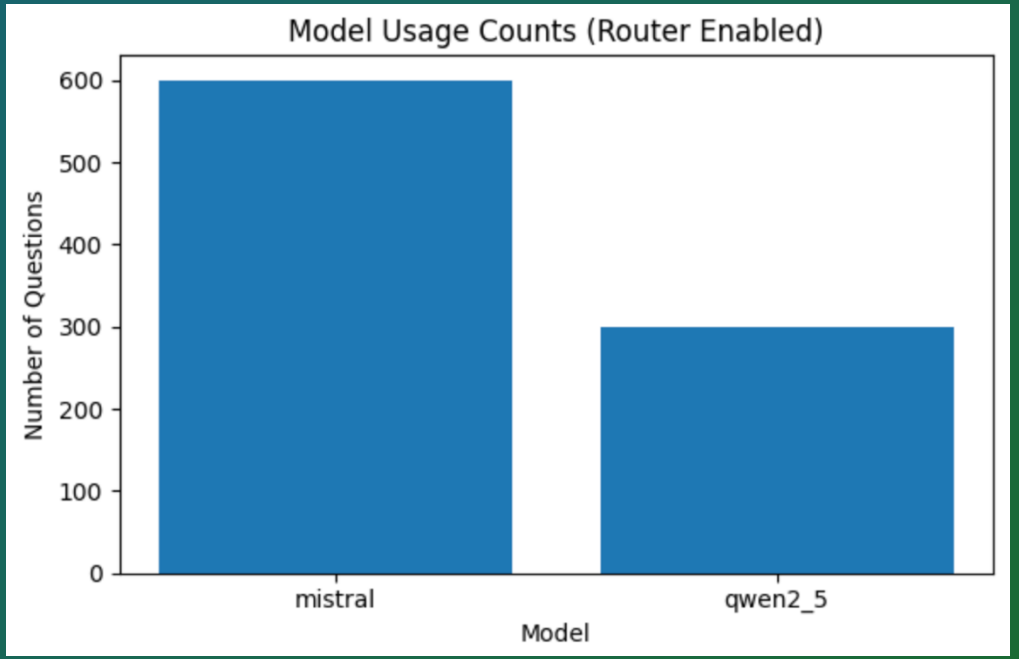



```
rows: 45000
unique questions: 15000
datasets: 3
models: ['llama3', 'mistral', 'qwen2_5']

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
```

Client Layer – Inference Container

Dataset Inference Routing Policy Results

- 900 sample questions representing all datasets equally
- Randomly mixed and sent to the router



	dataset	routed_variant	mean_cosine	mean_latency	count
0	hotpotqa	qwen2_5	0.655734	324.396267	300
1	narrativeqa	mistral	0.353230	365.782356	300
2	pubmedqa	mistral	0.236548	148.214912	300

ML Based Routing


```
================================================================================
Target label summary (best model per question)
================================================================================
mistral    10094
llama3      3254
qwen2_5     1652

================================================================================
Test results
================================================================================
Accuracy:          0.4673
Balanced accuracy: 0.4631

Classification report:
              precision    recall  f1-score   support

      llama3       0.34      0.63      0.44       651
     mistral       0.84      0.44      0.58      2019
     qwen2_5       0.15      0.32      0.20       330

    accuracy                           0.47      3000
   macro avg       0.44      0.46      0.41      3000
weighted avg       0.65      0.47      0.51      3000


Confusion matrix:
              pred:llama3  pred:mistral  pred:qwen2_5
true:llama3           407           55           189
true:mistral          699           888          432
true:qwen2_5          108           115          107
```

Features

Dataset, Prompt Characters, Context Characters

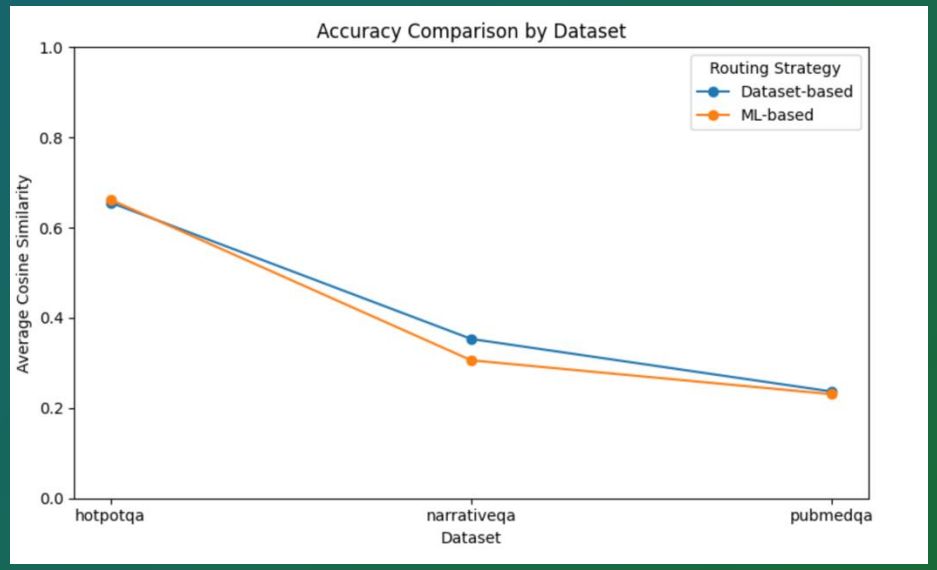
Target (Utility = best model for question)

$$\text{Utility} = \text{answer\_cosine\_sim} + w_{\text{em}} \cdot \text{exact\_match} - w_{\text{lat}} \cdot \text{inference\_ms}$$

Client Layer – Inference Container

Dataset Inference Routing Policy Results

- ML based routing policy showed similar results for hotpotqa and pubmedqa datasets
- Dataset routing policy did well for narrativeqa dataset



	dataset	routed_variant	mean_cosine	em_rate	mean_latency	count
0	hotpotqa	llama3	0.751629	0.571429	293.721953	77
1		mistral	0.870826	0.678571	209.791521	28
2		qwen2_5	0.597160	0.338462	346.132284	195
3	narrativeqa	llama3	0.305930	0.000000	371.175908	300
4	pubmedqa	mistral	0.240653	0.000000	152.463974	288
5		qwen2_5	-0.008202	0.000000	170.616461	12

Live Router Demo

http://129.114.26.184:8081/

Where We're Going

Data Layer

8 QA datasets

Unified schema

Available now on HuggingFace

TechQueen24/ DoxplainQA

Backend

3rd gen pipeline

AWS hosting & services integration

Automated drift detection & model selection

Frontend

3rd gen UI

React.js & Tailwind CSS

Production grade integration documentation

DoxplainQA Unified Schema

Dataset	Split	ID	Question	Context	Answer
Type: str	Type: str	Type: str	Type: str	Type: str	Type: str
Name of the originating dataset	Original train/test/ Validation split tag	Original dataset-specific identifier	Natural language question	Canonical answer string	Supporting textual evidence for answer

Current Datasets

boolq, drop, hotpotqa, narrativeqa, natural_questions, qasper, squad_v2, triviaqa_wiki (**pubmed coming soon!)**

Hugging Face | Search models, datasets, users... | Models | Datasets | Spaces | Community | Docs | Pricing

Datasets: TechQueen24 / DoxplainQA | like 0

Tasks: Question Answering | Languages: English | ArXiv: arxiv:1905.10044 | arxiv:1903.00161 | arxiv:1809.09600 | + 4

Dataset card | Files and versions | xet | Community | Settings

Dataset Viewer | API | Embed | Duplicate | Data Studio

Split (3)
train

▶ The dataset viewer is not available for this split.

Downloads last month 22

Edit dataset card

DoxplainQA: A Unified Question–Answering Dataset

View the ingestion code repository here

Overview

DoxplainQA is a **unified question–answering (QA) dataset** constructed to support **systematic evaluation, comparison, and explanation** of QA models across heterogeneous source datasets. The dataset harmonizes multiple established QA benchmarks into a **single, normalized schema**, enabling **consistent training, inference, and evaluation** pipelines within the Doxplain framework.

Discussions & Conclusions

- Data-centric model routing remains in research phase
- Potential applications for addressing prompt drift in real time
- Better features needed to accelerate model convergence

- Enterprise grade SLAs most likely achievable in cloud services
- Broken documentation is a significant barrier to deployment
- Consultation with industry experts would be beneficial