Locally Hosted RAG *for* Research Economists

Structured Report Generation

Samuel Braun and Minchul Shin

Federal Reserve Bank of Philadelphia

December 2025

Disclaimer: The views expressed herein are solely those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of Philadelphia or the Federal Reserve System. Nothing in the material should be construed as an endorsement of any organization or its products or services. Any errors or omissions are the responsibility of the authors. No statements here should be treated as legal advice. This is informational without any representations or assurances on applicability to others.

ML Group at the Philadelphia Fed

Our mission: We aim to understand how new technologies are reshaping the economy and financial markets, and how they can support core functions of central banking.

Members:

- Minchul Shin, Economist
- Vitaly Meursault, Economist
- Simon Freyaldenhoven, Economist
- Ryan Kobler, Research Analyst
- Samuel Braun, Research Assistant

Our Journey Towards Local AI Systems

- **Original question:** Can AI replace parts of what economist do? Which parts, when, and how?
  - After ChatGPT-3.5 (Nov 2022), this became a practical question, not a hypothetical one.
  - First internal tool (Dec 2023): classified new public reports and generated standardized two-page summaries.

- Security and other considerations led us to locally hosted models.

From Static Reports to Interactive RAG

- We built **briefing helper**, an experimental dashboard powered by locally hosted LLMs (e.g., GPT-OSS 120B, Gemma 3 series) using public and other data.
  - It mostly produces **static structured reports**: text in, fixed summary out.

- But when economists (and AI systems) write and analyze, they need to interact with the underlying textual data, not just read summaries.

- This calls for a more **dynamic retrieval layer** that:
  - Manges context efficiently for LLMs.
  - Lets economists explore and quote from our text corpus on demand.

Goal for our current RAG system

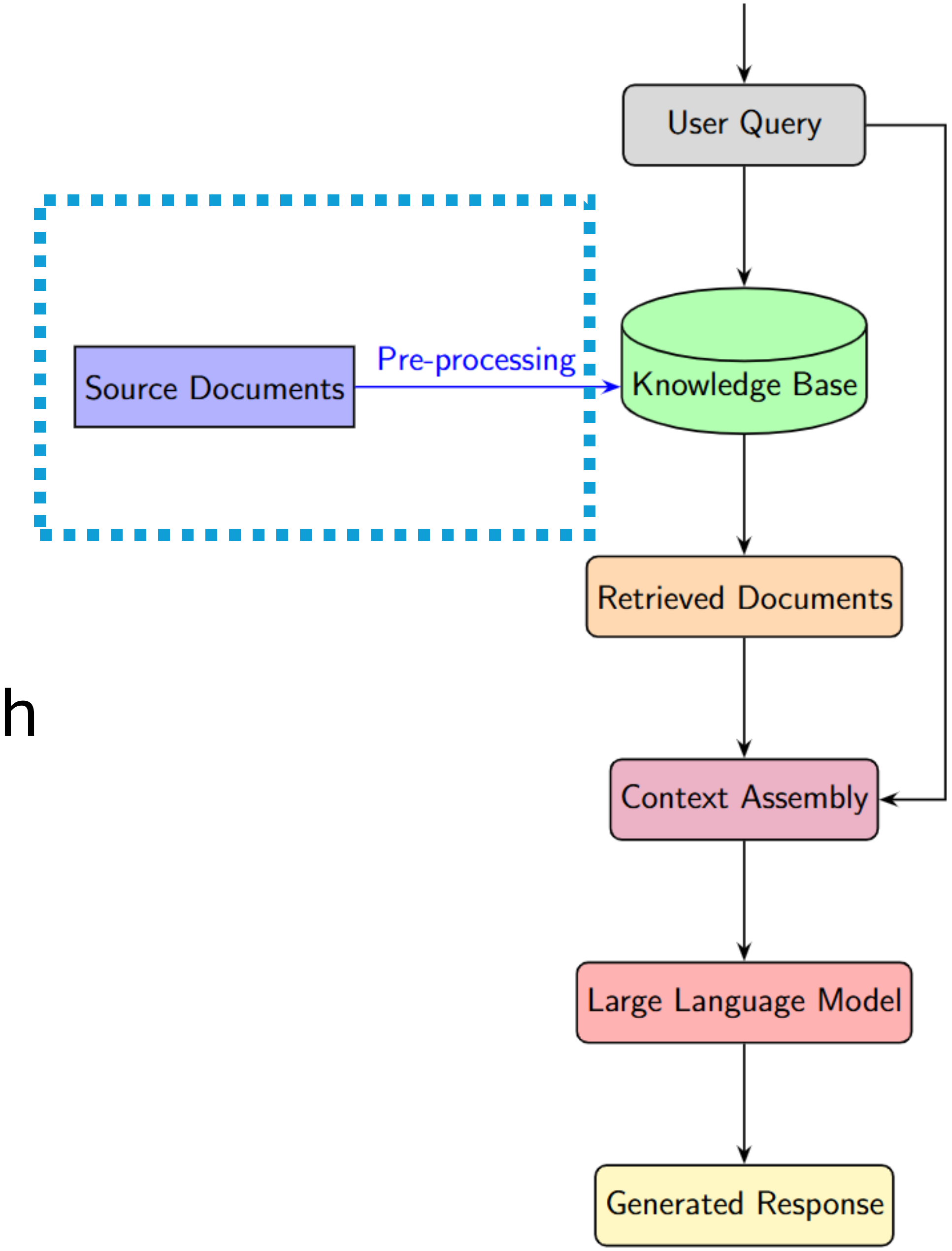
- The goal is to make a RAG system directly useful for what our economists do when they write memos and monitoring reports.

- We focus on low-creativity, fact-intensive writing.
    - To judge success, we reverse-engineer economists' past writing:
        - Take a sentence from an actual memo and turn it into a question.
        - Sentence: "In Q2, GDP grew 3.84%."
        - Question: "What was the reported Q2 GDP growth rate?"
    - An ideal RAG system should answer in a way that it is factually consistent with economists' own wording.

RAG Architecture

Data Pre-Processing

- **Input data**: News articles and data reports
- **Pre-processing**
  - Text cleaning and chunking
    - Chunks consist of ~150 tokens each
  - Chunk embeddings generated and inserted into **knowledge base**
- Updated on daily basis



RAG Architecture

Data Pre-Processing: Enhanced Chunking

- Two components added to chunks:
  1. Prepend publication date
  2. Prepend chunk-specific context: **context prefix**
     - Inspired by research from Anthropic (2024)
     - Generated using Gemma 3 (27b)
- Chunk format:

```
"""{source publication date}
   {context prefix}
   {source text chunk}"""
```

```
# simplified context generation query

context_gen_query = """Here is a document that you are
chunking and will eventually embed for a Retrieval-Augmented
Generation (RAG) agent:
<document>
{full_text}
</document>

Here is the chunk we want to situate within the whole
document:
<chunk>
{chunk}
</chunk>

Your goal is to create contextual information for this chunk
to improve embedding search retrieval. Expand any acronyms
listed in the chunk. If the subject of the chunk is not
included in the chunk, state the complete subject of the
chunk. If the chunk includes a quote but the speaker of the
quote is not included, state the speaker of the quote."""
```

RAG Architecture

Data Pre-Processing: Example Chunk

Original chunk:

"From historic investments in site development to cutting red tape and recruiting companies to move and grow here, our strategy is working. We’ve brought in billions of dollars in private sector investment, created thousands of good-paying jobs, and made Pennsylvania one of the best places in the country to live, work, and build a business."

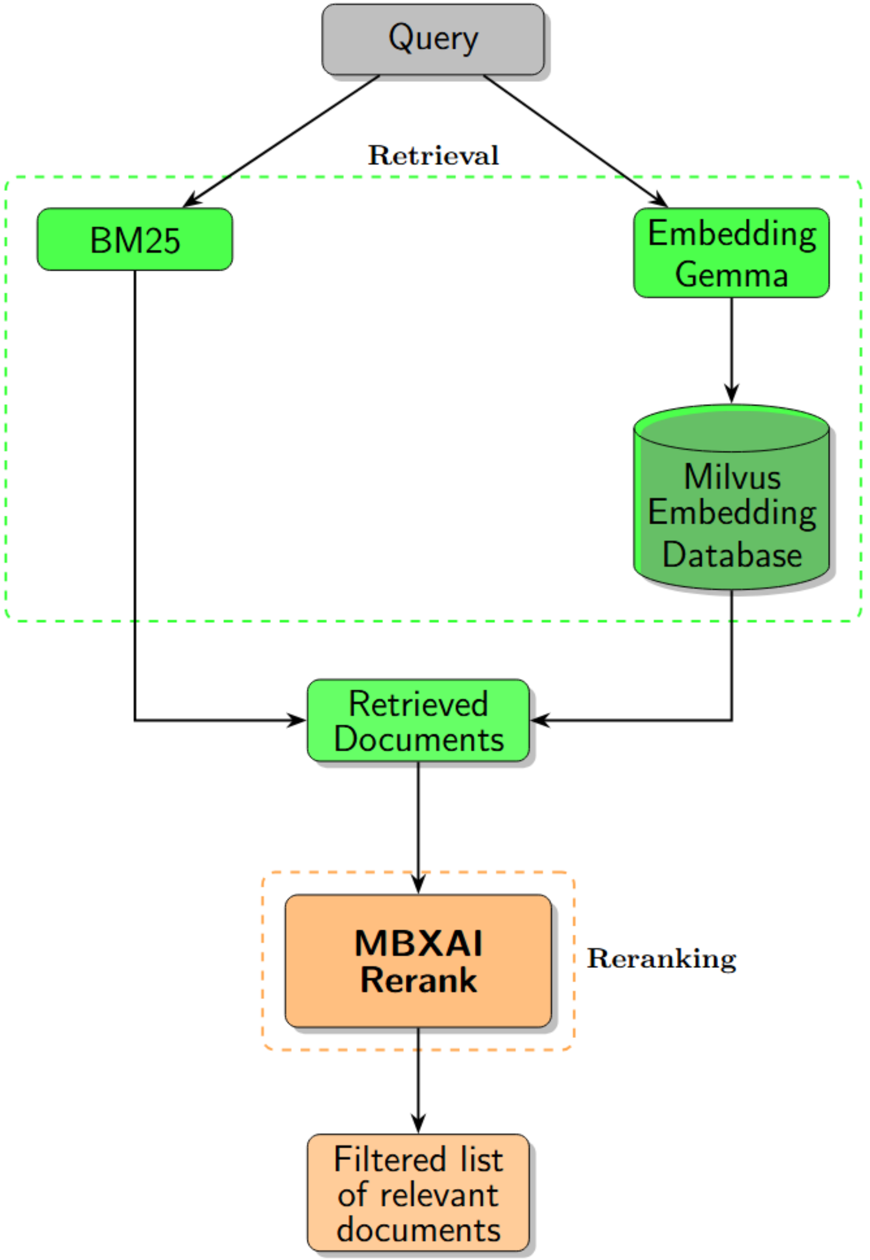
Chunk with context prefix and metadata:

published date: October 28, 2025
context: This chunk is spoken by Gov. Josh Shapiro regarding Pennsylvania’s economic strategy. New data from the Pennsylvania Department of Community and Economic Development (DCED) reveals that more than $31.6 billion has been secured in private sector investments and over 16,700 jobs created since the start of the Shapiro Administration, demonstrating positive economic growth across 16 of 20 industry sectors. article: "From historic investments in site development to cutting red tape and recruiting companies to move and grow here, our strategy is working. We’ve brought in billions of dollars in private sector investment, created thousands of good-paying jobs, and made Pennsylvania one of the best places in the country to live, work, and build a business."

RAG Architecture

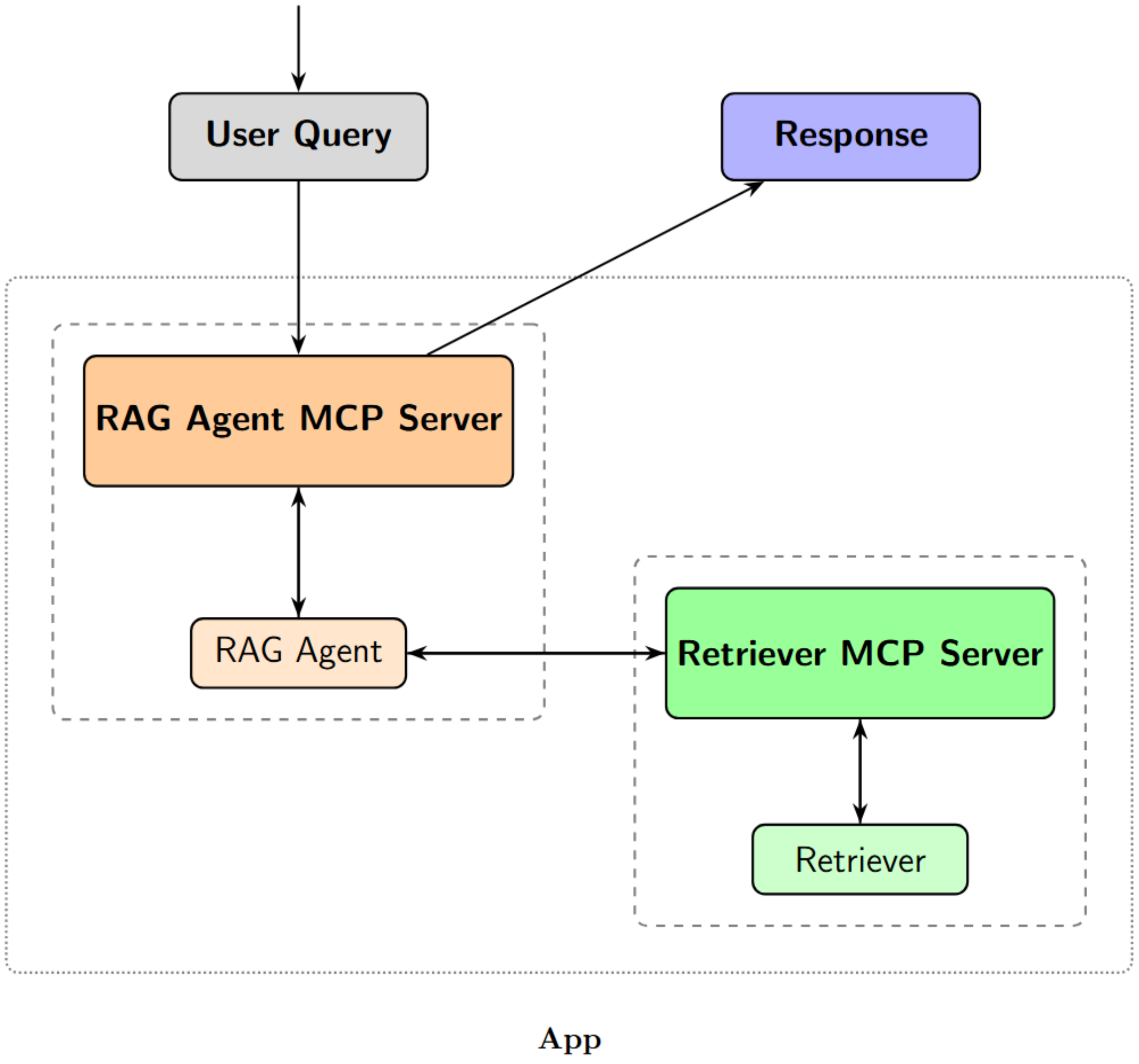
Retriever

- Two main stages: retrieval and reranking
- **Ensemble retriever**
  1. Semantic search (weight: 50%)
     - Model: EmbeddingGemma
     - Database: Milvus
     - **Document count**: 50 documents
  2. Keyword search (BM25) (weight: 50%)
     - BM25 searches on discrete text items
     - **Document count**: 50 documents
- **Reranker**
  - MBXAI Rerank Large V2
  - Cross-encoder: process query and document together
  - **Document count**: 15 documents



RAG Architecture
MCP Design

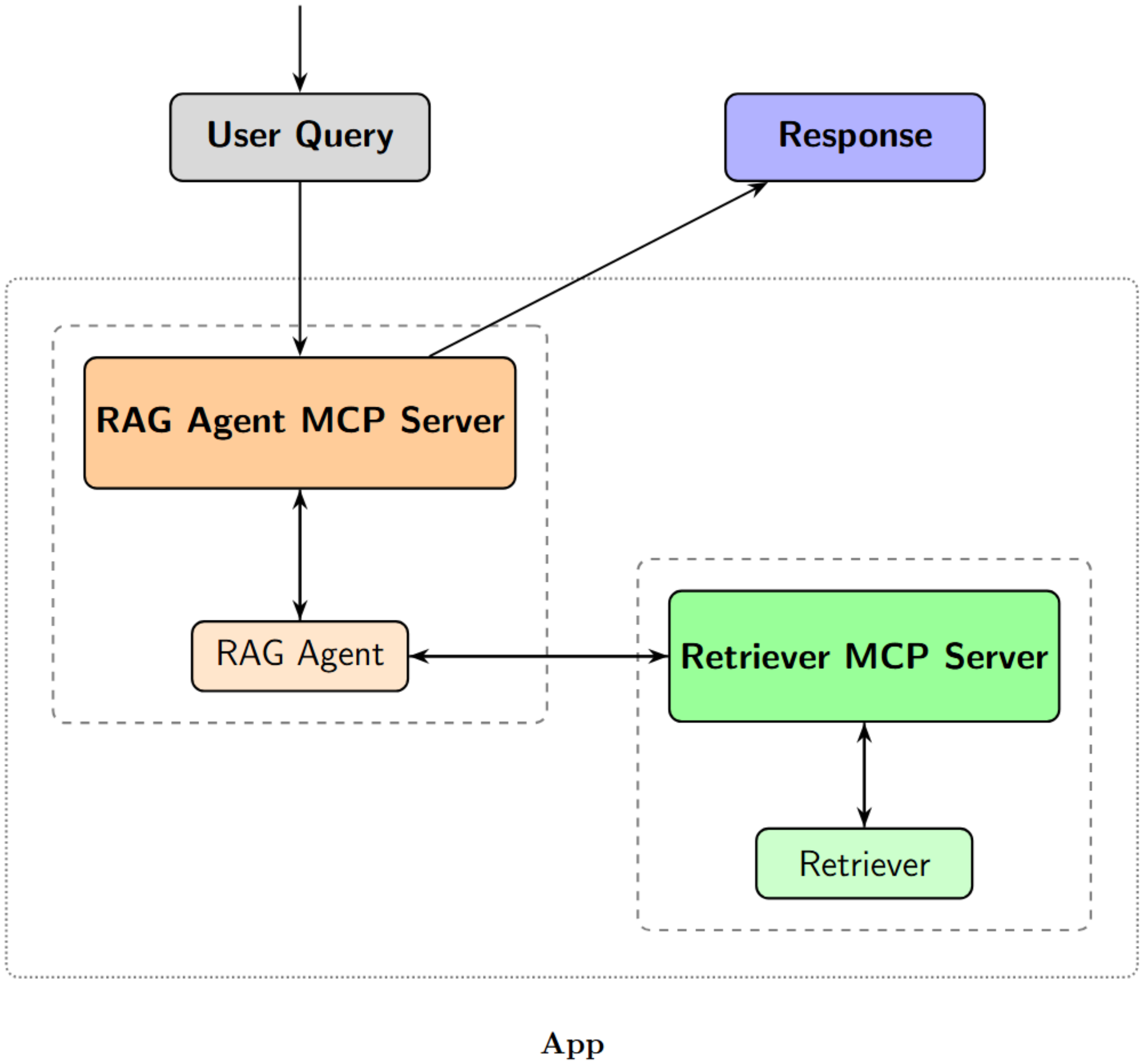
- Model Context Protocol (MCP)
  - Open standard for connecting AI applications to tools and data sources
  - Wraps functions as servable APIs
- **RAG agent MCP**
  - Input: User query
  - Output: Generated answer
- **Retriever MCP**
  - Input: User query
  - Output: Documents relevant to user query



RAG Architecture

MCP Design: Workflow

1. User submits query to app
2. App calls RAG MCP with query
3. RAG MCP calls Retriever MCP
4. Retriever MCP returns documents to RAG MCP
5. RAG MCP generates response
6. Response returned to app
7. App returns response to user



RAG Architecture
MCP Design: Motivation

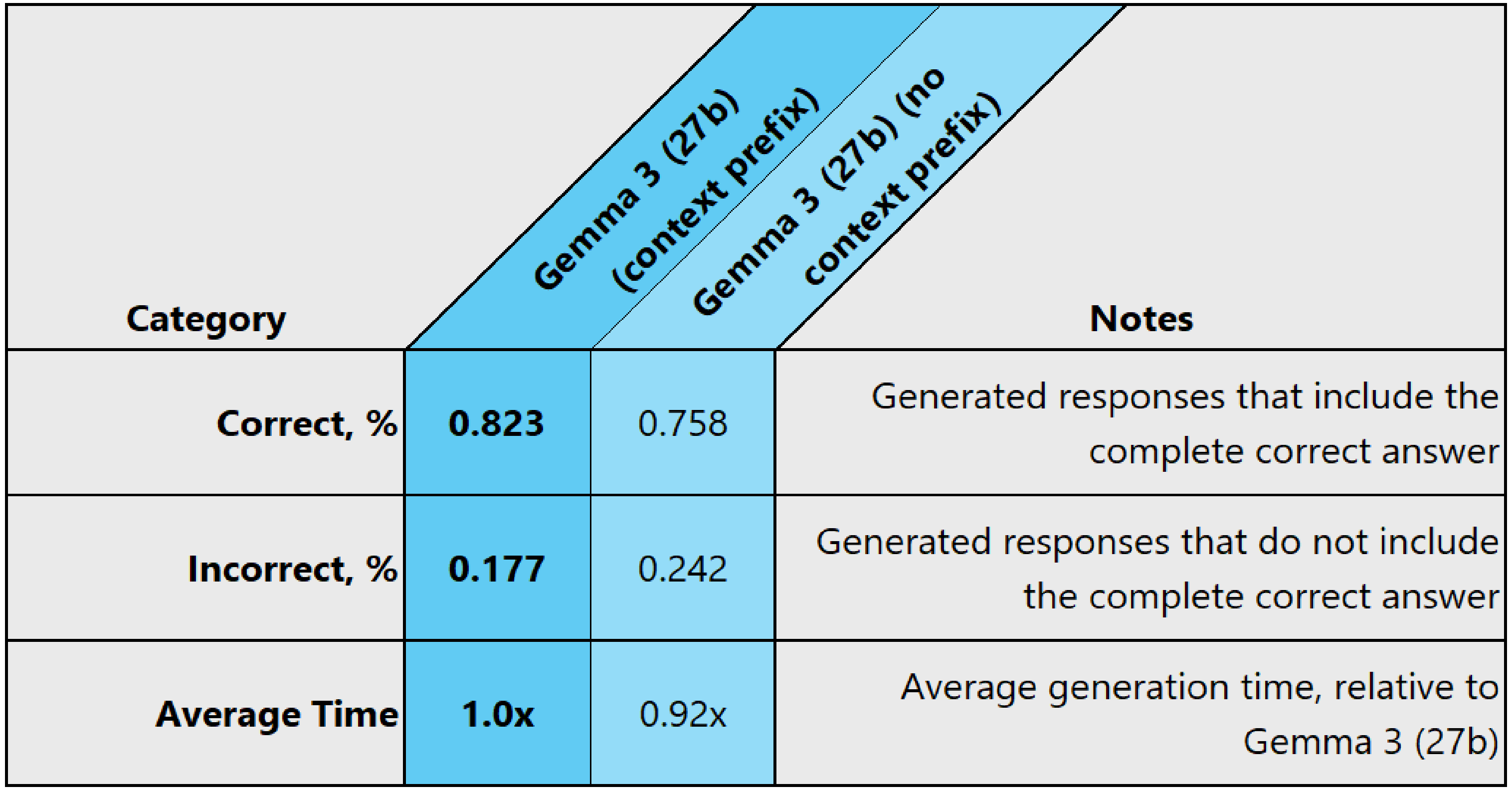
- Foundational block in multi-agent framework
  - Wrapping our RAG agent transforms it into a callable tool
    - **RAG MCP**: Multiple use cases can generate answers from the same centralized knowledge base
      - Use Case #1: Tool called by report generation application
      - Use Case #2: Backbone of economist Q&A agent
  - RAG agent and retriever MCPs act as local "sub-agent(s)"

Evaluation
Evaluation Methodology

- Test Dataset #1: Q&A pairs adapted from sentences actual economists have written, 62 pairs
  - **Goal**: Test coverage of our database on information reflective of specific Philadelphia Fed economist use cases
  - Reverse-engineered economists' past writing:
    - Take sentences from actual memos and turn them into questions
    - "In Q2, GDP grew 3.84%." → "What was the reported Q2 GDP growth rate?"
- Test Dataset #2: synthetically generated Q&A pairs, 1000 pairs
  - Generated from documents in RAG knowledge base
  - **Goal**: Evaluate general RAG performance by testing retriever and generation

Evaluation

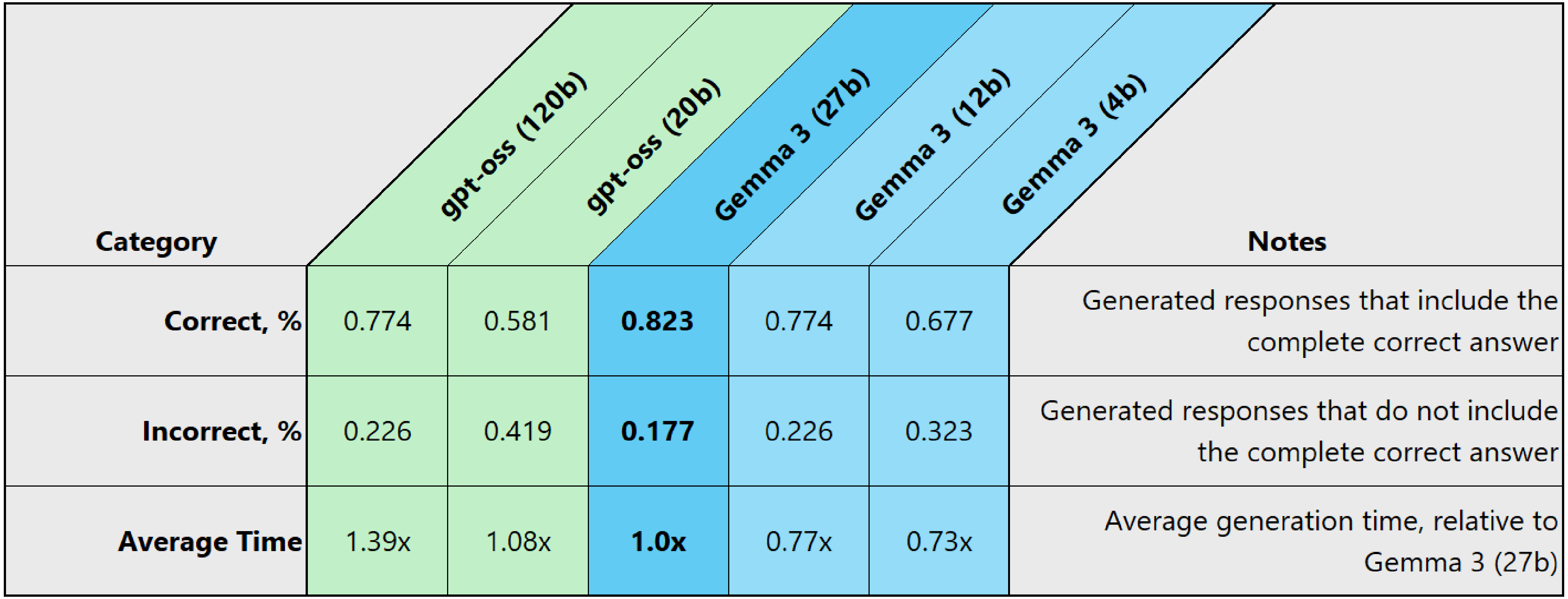
Dataset #1: Economist Workflows



Category	Gemma 3 (27b) (context prefix)	Gemma 3 (27b) (no context prefix)	Notes
Correct, %	**0.823**	0.758	Generated responses that include the complete correct answer
Incorrect, %	**0.177**	0.242	Generated responses that do not include the complete correct answer
Average Time	**1.0x**	0.92x	Average generation time, relative to Gemma 3 (27b)

Evaluation

Dataset #1: Economist Workflows

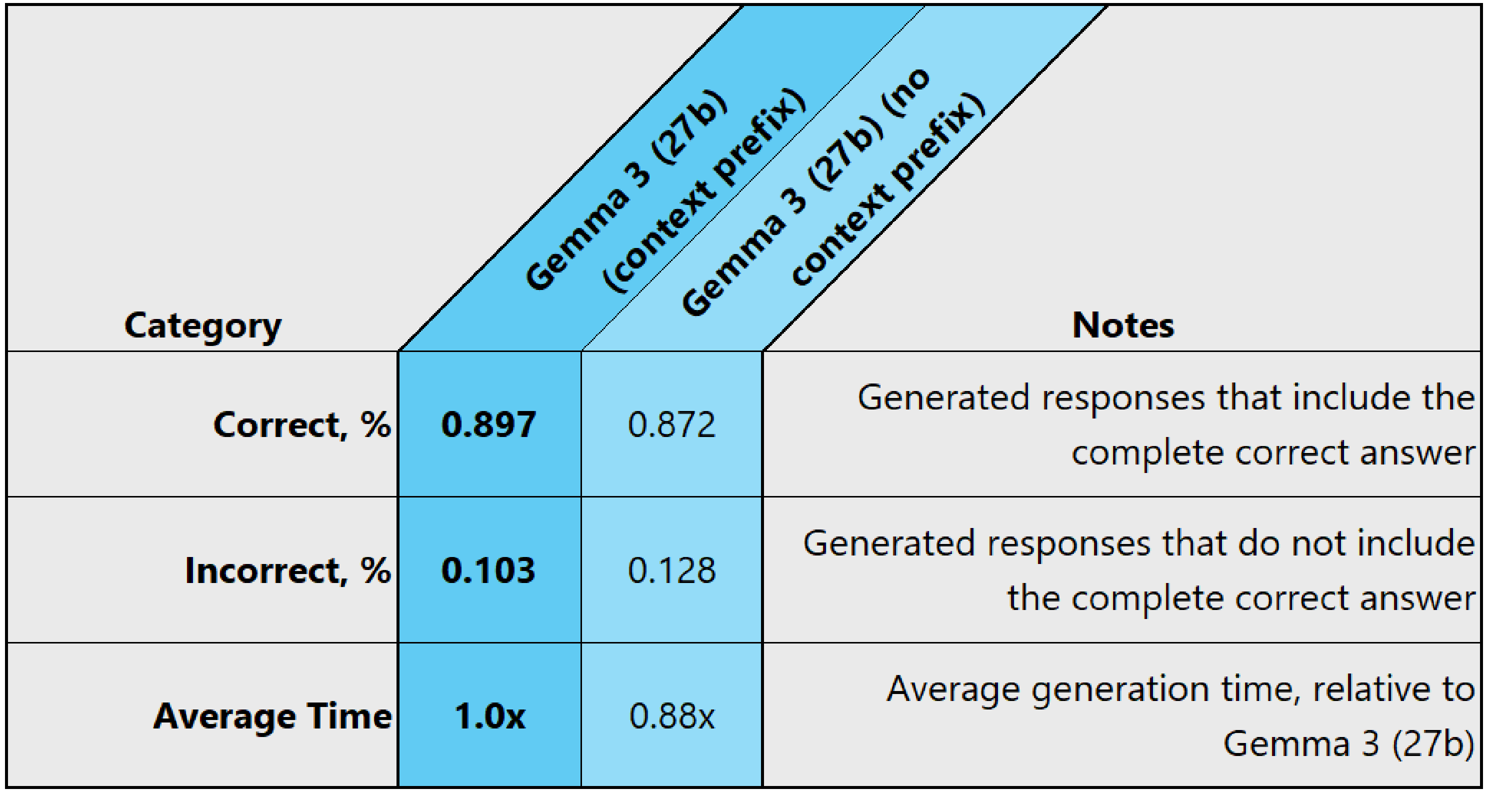


Category	gpt-oss (120b)	gpt-oss (20b)	Gemma 3 (27b)	Gemma 3 (12b)	Gemma 3 (4b)	Notes
Correct, %	0.774	0.581	**0.823**	0.774	0.677	Generated responses that include the complete correct answer
Incorrect, %	0.226	0.419	**0.177**	0.226	0.323	Generated responses that do not include the complete correct answer
Average Time	1.39x	1.08x	**1.0x**	0.77x	0.73x	Average generation time, relative to Gemma 3 (27b)

Above evaluations conducted with context prefix-enabled retriever.

Evaluation

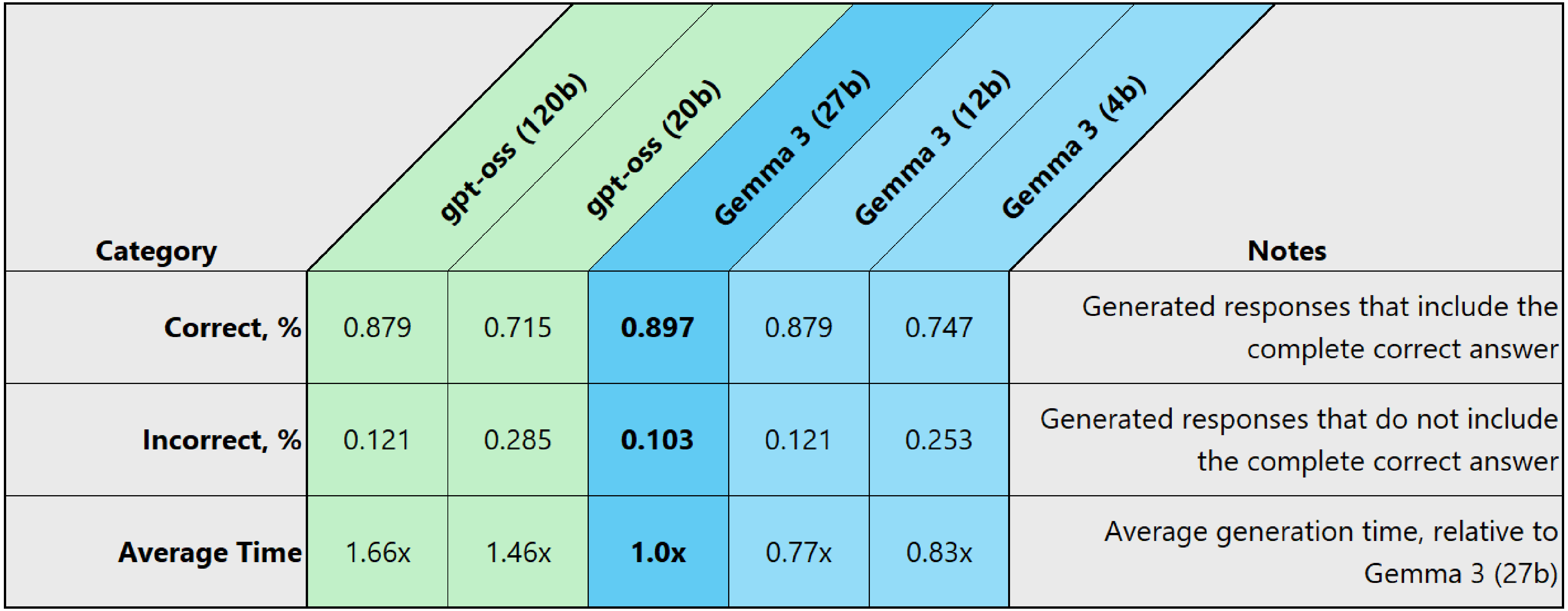
Dataset #2: Basic RAG Agent Performance



Category	Gemma 3 (27b) (context prefix)	Gemma 3 (27b) (no context prefix)	Notes
Correct, %	**0.897**	0.872	Generated responses that include the complete correct answer
Incorrect, %	**0.103**	0.128	Generated responses that do not include the complete correct answer
Average Time	**1.0x**	0.88x	Average generation time, relative to Gemma 3 (27b)

Evaluation

Dataset #2: Basic RAG Agent Performance



Category	gpt-oss (120b)	gpt-oss (20b)	Gemma 3 (27b)	Gemma 3 (12b)	Gemma 3 (4b)	Notes
Correct, %	0.879	0.715	**0.897**	0.879	0.747	Generated responses that include the complete correct answer
Incorrect, %	0.121	0.285	**0.103**	0.121	0.253	Generated responses that do not include the complete correct answer
Average Time	1.66x	1.46x	**1.0x**	0.77x	0.83x	Average generation time, relative to Gemma 3 (27b)

Above evaluations conducted with context prefix-enabled retriever.

Future Development
Use Concepts

- Currently in test stage
  - Economists testing basic RAG Agent functionality
- Future applications
  - Strength of system: our up-to-date database for research economists
  - Use cases in citation verification and correction
  - Local deep research report generation
  - Expanded database: Information on the Third District

Q&A