Assignment 2

Opened: Sunday, 27 July 2025, 12:00 AM
Due: Sunday, 24 August 2025, 11:59 PM

Assignment 2 – Comparative Financial QA System: RAG vs Fine-Tuning

Objective

Develop and compare two systems for answering questions based on company financial statements (last two years):

1. Retrieval-Augmented Generation (RAG) Chatbot: Combines document retrieval and generative response.
2. Fine-Tuned Language Model (FT) Chatbot: Directly fine-tunes a small open-source language model on financial Q&A.

Use the same financial data for both methods and perform a detailed comparison on accuracy, speed, and robustness.

Step-by-Step Tasks

1. Data Collection & Preprocessing

- Obtain financial statements for the last two years (publicly available or from a group member's company).
- Convert documents (PDF, Excel, HTML) to plain text using OCR or appropriate parsers.
- Clean text by removing noise like headers, footers, and page numbers.
- Segment reports into logical sections (e.g., income statement, balance sheet).
- Construct at least 50 question-answer (Q/A) pairs reflecting the financial data.
    - Example:
        - Q: What was the company's revenue in 2023?
        - A: The company's revenue in 2023 was $4.13 billion.

2. Retrieval-Augmented Generation (RAG) System Implementation

2.1 Data Processing

- Split the cleaned text into chunks suitable for retrieval with at least two chunk sizes (e.g., 100 and 400 tokens).
- Assign unique IDs and metadata to chunks.

2.2 Embedding & Indexing

- Embed chunks using a small open-source sentence embedding model (e.g., all-MiniLM-L6-v2, E5-small-v2).
- Build:
    - Dense vector store (e.g., FAISS, ChromaDB).
    - Sparse index (BM25 or TF-IDF) for keyword retrieval.

2.3 Hybrid Retrieval Pipeline

- For each user query:
    - Preprocess (clean, lowercase, stopword removal).
    - Generate query embedding.
    - Retrieve top-N chunks from:
        - Dense retrieval (vector similarity).
        - Sparse retrieval (BM25).
    - Combine results by union or weighted score fusion.

2.4 Advanced RAG Technique (Select One)

Remainder (Group Number mod 5)	Advanced Technique	Description
1	Multi-Stage Retrieval	Stage 1: Broad retrieval; Stage 2: Re-rank candidates using a precise cross-encoder model.
2	Chunk Merging & Adaptive Retrieval	Dynamically merge adjacent chunks or adapt chunk size based on query complexity or length.
3	Re-Ranking with Cross-Encoders	Use a cross-encoder to re-rank top retrieved chunks based on query relevance.
4	Hybrid Search (Sparse + Dense Retrieval)	Combine BM25 keyword search with dense vector retrieval for balanced recall and precision.
0	Memory-Augmented Retrieval	Supplement retrieval with a persistent memory bank of frequently asked or important Q&A pairs.

- Implement and document your assigned technique in detail.

2.5 Response Generation

- Use a small, open-source generative model (e.g., DistilGPT2, GPT-2 Small, Llama-2 7B if available).
- Concatenate retrieved passages and user query as input to generate the final answer.
- Limit total input tokens to the model context window.

2.6 Guardrail Implementation

- Implement one guardrail:
  - Input-side: Validate queries to filter out irrelevant or harmful inputs.
  - Output-side: Filter or flag hallucinated or non-factual outputs.

2.7 Interface Development

- Build a user interface (Streamlit, Gradio, CLI, or GUI).
- Features:
  - Accept user query.
  - Display answer, retrieval confidence score, method used, and response time.
  - Allow switching between RAG and Fine-Tuned modes.

3. Fine-Tuned Model System Implementation

3.1 Q/A Dataset Preparation

- Use the same ~50 Q/A pairs as for RAG but convert into a fine-tuning dataset format.

3.2 Model Selection

- Choose a small open-source language model suitable for fine-tuning:
  - Examples: DistilBERT, MiniLM, GPT-2 Small/Medium, Llama-2 7B, Falcon 7B, Mistral 7B.
- Ensure no use of closed or proprietary APIs.

3.3 Baseline Benchmarking (Pre-Fine-Tuning)

- Evaluate the pre-trained base model on at least 10 test questions.
- Record accuracy, confidence (if available), and inference speed.

3.4 Fine-Tuning

- Fine-tune the selected model on your Q/A dataset.
- Log all hyperparameters:
    - Learning rate, batch size, number of epochs, compute setup (CPU/GPU).
- Use efficient techniques as assigned (see next).

3.5 Advanced Fine-Tuning Technique (Select One)

Remainder (Group Number mod 5)	Advanced Fine-Tuning Technique	Description
1	Supervised Instruction Fine-Tuning	Fine-tune on instruction-style Q/A pairs using supervised learning.
2	Adapter-Based Parameter-Efficient Tuning	Tune small adapter modules inserted into base model to reduce training cost.
3	Mixture-of-Experts Fine-Tuning	Use multi-expert architectures for efficient fine-tuning and inference.
4	Retrieval-Augmented Fine-Tuning	Combine retrieval mechanisms with fine-tuning for improved knowledge grounding.
0	Continual Learning / Domain Adaptation	Fine-tune incrementally on new financial data while preserving prior knowledge.

- Implement and document the advanced fine-tuning method in the notebook.

3.6 Guardrail Implementation

- Implement one guardrail (input or output side, similar to RAG).

3.7 Interface Development

- Integrate fine-tuned model into the same UI as RAG.
- Show:
    - Answer, confidence score, method name, inference time.
    - Ability to switch between RAG and fine-tuned model.

4. Testing, Evaluation & Comparison

4.1 Test Questions (Mandatory)

For both systems, ask three official questions:

1. Relevant, high-confidence: Clear fact in data.
2. Relevant, low-confidence: Ambiguous or sparse information.
3. Irrelevant: Example: "What is the capital of France?"

4.2 Extended Evaluation

- Evaluate both systems on at least 10 different financial questions.
- For each system and question, record:
    - Real (ground-truth) answer
    - Model-generated answer
    - Confidence score (or probability if available)
    - Response time (seconds)
    - Correctness (Y/N)

4.3 Results Table Example

Question	Method	Answer	Confidence	Time (s)	Correct (Y/N)
Revenue in 2023?	RAG	$4.02B	0.92	0.50	Y
Revenue in 2023?	Fine-Tune	$4.13B	0.93	0.41	Y
Unique products?	RAG	13,000 units	0.81	0.79	Y
Unique products?	Fine-Tune	13,240 units	0.89	0.65	Y
Capital of France?	RAG	Data not in scope	0.35	0.46	Y
Capital of France?	Fine-Tune	Not applicable	0.85	0.38	Y

4.4 Analysis

- Compare average inference speed and accuracy.
- Discuss:
    - Strengths of RAG (e.g., adaptability, factual grounding).
    - Strengths of Fine-Tuning (e.g., fluency, efficiency).

  - Robustness to irrelevant queries.
  - Practical trade-offs.

5. Submission Requirements

- Submit one ZIP file per group, with naming convention: Group_<Number>_RAG_vs_FT.zip.
- ZIP must include:
  - Python Notebook (.ipynb or .py) containing:
    - Data processing steps.
    - Both RAG and fine-tuning implementations with markdown explanations.
    - Advanced technique section for RAG and fine-tuning.
    - Testing and comparison tables.
  - PDF report with:
    - 3 screenshots showing test queries, answers, confidence scores, inference times, and method used.
    - Summary comparison table.
    - Hosted app link (Streamlit/Gradio/etc.) for demonstration.
- Use only open-source models and software; no proprietary APIs.
- Clearly comment and document all code and steps.

Notes & Recommendations

- Feel free to use free or institutional GPU resources (Google Colab, Kaggle, campus clusters).
- The quantitative comparison and detailed documentation of both methods are critical for grading.
- Implementing clear guardrails is mandatory for responsible AI.
- The UI should be user-friendly and clearly indicate which method is producing the answer.

This assignment aims to give you thorough hands-on experience in building and comparing two major retrieval and generation paradigms for specialized financial question answering with open-source technologies. Good luck!

Add submission

Submission status

Group	Group 16
Attempt number	This is attempt 1 (3 attempts allowed).
Submission status	Nothing has been submitted for this assignment
Grading status	Not graded
Time remaining	16 days 14 hours remaining
Last modified	-

← Previous
Assignment 1

Next →
Webinar-1