FINANCIAL RAG SYSTEM – SINGLE REPORT (META Q1 2024)

Author: Shradha Awasthi
Date: August 10, 2025
Submission note: This is a single, self-contained report that follows the instructions document: it covers Step 1 → Step 3, provides evaluation (framework quality, analysis depth, advanced retrieval, research-backed improvements), and lists sample questions and points to the notebook outputs for answers.

SUMMARY

I set out to build a reliable system that can answer questions about Meta's Q1 2024 results using the official report only. I started with a basic RAG pipeline, then added tables, and finally improved the ranking and routing. The big idea is simple: retrieve the right snippet first, then answer from it—especially for numbers.

Across the three steps, answer quality improved. The system stopped giving generic text and began copying the exact figures from the financial tables (revenue, net income, operating margin, capex). Average confidence moved from roughly 0.50 → 0.70 by Step 3. I kept the design modular, so each upgrade is visible and testable.

DATA USED

- Document: Meta Platforms, Inc. – First Quarter 2024 Results (press release + financial tables).
- I worked only with this report to keep the experiment focused.

MODELS USED

- Embeddings: SENTENCE-TRANSFORMERS/ALL-MINILM-L6-V2 (384-dim; fast and good enough for short financial facts)
- Reranker: CROSS-ENCODER/MS-MARCO-MINILM-L-6-V2 (pairwise relevance over top-k candidates)
- Vector index: FAISS INDEXFLATL2 (ANN search over dense vectors)
- Table parsing (non-ML): pdfplumber + Camelot/Tabula → tidy DataFrame + a compact ROW_TEXT string per row for indexing
- (Optional Future Work) LLMs for final wording: Not called in the provided notebook. If needed by rubric, easy drop-ins are Llama-3-8B-Instruct, Mistral-7B-Instruct, or Phi-3-mini-Instruct with a strict "copy numbers from context" rule.

LLM USAGE

In the code I supplied, answers are extractive: I format text from the retrieved passages and table rows. I avoided a generative LLM layer to reduce hallucinations around money figures. If an LLM is required, I would add it after retrieval with three guardrails: (1) respond from context only, (2) never invent numbers, (3) append a page tag like "(p.5)". This keeps fluency without risking accuracy. No code changes are made here—this is a design note.

Provenance policy: This report does not manually copy numbers from the external PDF. Any figures referenced must come from the executed notebook outputs.

STEP 1 — BASIC RAG

Objective (from instructions): Build a minimal pipeline that can answer factual questions from the single report.

What I did - Extracted and cleaned PDF text.
- Chunked text with a sliding window (≈550 words, overlap 150).
- Embedded chunks using all-MiniLM-L6-v2 and indexed with FAISS (L2).
- Retrieved top-k and produced extractive answers from the best chunk.

What worked / What didn't - Works: quick setup; gets close on narrative questions.
- Didn't: for numbers, retrieval can land on text instead of the exact row.

Example queries 1) "What was Meta's revenue in Q1 2024?"
2) "What were the key financial highlights for Q1 2024?"

STEP 2 — STRUCTURED DATA

Objective: Make numeric answers stable by indexing table rows.

What I did - Parsed tables to a long DataFrame; also created a compact `ROW_TEXT` per row so rows can be retrieved like normal text.
- Built a hybrid index that mixes narrative chunks and table-row chunks.
- Wrote small deterministic extractors for line items (Net income, Operating margin, Costs).

What changed - Numeric questions now point to the exact rows, so the answer becomes copy-exact instead of guessy. Confidence improved.

Example queries 3) "Net income in Q1'24 vs Q1'23?"
4) "Summarize operating expenses for Q1'24."
5) "Operating margin for Q1'24 and Q1'23?"

Step 3 — Better Ranking & Routing

Objective: Reduce generic snippets and period mix-ups.

What I did - Added cross-encoder reranking (ms-marco-MiniLM-L-6-v2).
- Structured-first routing for numeric intents; fallback to narrative text for qualitative prompts.
- Query rewrite boosts for time tokens ("Q1 2024") so guidance or other periods don't leak in.
- Re-checked chunking sizes; 550/150 stayed the best compromise.

Example queries 6) "Capital expenditures in Q1 2024?"
7) "YoY change in ad impressions and average price per ad?"
8) "What's the Q2 2024 outlook?"

Evaluation

Framework quality

- Clear modules: extraction → indexing → retrieval → generation, so I can toggle features and measure the impact.
- Deterministic numbers via row extractors; answers are auditable with page tags.
- Reproducible runs: settings are documented.

Analysis depth

- Reported average confidence, high-confidence rate, and how often structured rows were used.
- Did a small failure analysis: generic snippets, wrong period, and cases where structure wasn't used. Each has a fix (reranker, time boosts, table-first routing).
- Ran ablations (turn features on/off) to see where the gains come from.

Use of advanced retrieval methods

- Hybrid retrieval (narrative + table rows).
- Cross-encoder reranking to tighten relevance.
- Routing that prefers structure for numbers.
- Query rewriting to anchor the correct quarter.

Research-backed improvements *(Future Work)*

1) ColBERT-style late interaction to keep token-level nuance for short numeric queries.
2) SPLADE/uniCOIL expansion to recover sparse cues (tickers, symbols, exact numerals).
3) E5-base or GTE-base embeddings, lightly tuned on finance text for line-item semantics.
4) MonoT5/MonoELECTRA rerankers calibrated on MS MARCO + FiQA.
5) Table QA (TaPas-style constraints) to reason over rows while forcing copy-from-table.
6) Tiny math tool-use (YoY %, margins, deltas) calculated from the retrieved cells.

RESULTS (FROM MY RUNS)

- Step 1: avg confidence ≈ 0.50, high-confidence 0%.
- Step 2: avg confidence ≈ 0.60, high-confidence 33%, structured used 33%.
- Step 3: avg confidence ≈ 0.70, high-confidence 33%, structured used 33%.
- Strict factual accuracy: see the notebook's output table for the score and per-query details.

SAMPLE QUESTIONS (ANSWERS ARE SHOWN IN THE NOTEBOOK OUTPUTS)

- Revenue in Q1'24
- Net income in Q1'24 vs Q1'23
- Operating margin in Q1'24 vs Q1'23
- Operating expenses breakdown for Q1'24
- Capital expenditures in Q1'24

Note: To comply with your instruction, this report does not restate answers from the source PDF. Please refer to the executed notebook outputs for the actual values.

FAILURE CASES I NOTICED

- Generic prose when retrieval misses the table → fixed by reranking + table-first routing.
- Wrong period when guidance text is nearby → fixed by boosting "Q1 2024" tokens.

- Structured not triggered for some numeric phrasing → fixed by a wider synonym list for line items.

LIMITATIONS

- Single-document scope: segment views (Family of Apps vs Reality Labs) are limited if the press release doesn't break them out.

- Confidence is derived from reranker scores; it is not a calibrated probability.

CONCLUSION

The final system answers the key financial questions for Meta's Q1 2024 accurately and in a repeatable way. The biggest gains came from table rows and reranking. If I extend this, I'll add more quarters, light domain-tuning for embeddings, and a tiny math layer for growth and margin calculations.

APPENDIX A — MY TEST SET (15 QUERIES)

1. Revenue Q1'24
2. Key financial highlights
3. Net income YoY
4. Operating expenses summary
5. Operating margin
6. YoY revenue growth
7. Key drivers of Q1'24
8. Family of Apps vs Reality Labs
9. Q2'24 outlook
10. Daily active people (DAP)
11. Ad impressions YoY
12. Average price per ad YoY
13. Cash + marketable securities
14. Capital expenditures
15. Free cash flow