Architecture & Agent Design Report

Overview

This prototype implements an end-to-end text analytics pipeline on Google Cloud, producing entity extraction, sentiment analysis, and concise summaries for each row in a CSV dataset. It also includes a lightweight agentic workflow that retrieves the most relevant rows to a user query, analyzes them with tool calls, and synthesizes a final answer.

The design emphasizes fast setup, resilience (graceful fallbacks when cloud access is unavailable), and clarity of outputs. It runs locally against a sample dataset and can optionally use Cloud Storage (gs://) and Vertex AI.

GCP Services Used

- Cloud Natural Language API (v2): entity and sentiment extraction.
- Vertex AI (optional fallback): Gemini models via Vertex when API key path isn't used.
- Google Generative AI (Gemini): summarization via API key.
- Cloud Storage (optional): load datasets from gs:// when configured.

Agentic Workflow

The agent composes three capabilities into a small chain:

- Retrieval: ranks rows by keyword overlap with the query; selects top-k candidates.
- Per-document tools: for each candidate, calls GCP Language (entities, sentiment) and Gemini (summary).
- Synthesis: concatenates per-document summaries and asks Gemini to produce a concise answer with context.
- Resilience: exceptions from any tool are captured; the chain continues with available signals.

Tool routing: if a `GOOGLE_API_KEY` is set, Gemini API is used directly; otherwise, if `USE_VERTEX_SUMMARY=true`, Vertex AI is used; else a dependency-free local summarizer provides deterministic behavior.

Results, Challenges, and Trade-offs

Outputs include `results.csv` (row-level entities, sentiment, summary), `eda.txt` (dataset stats), and a log. Keyword retrieval is simple and fast but less robust than embeddings; Language v2 entities may be generic (NUMBER/OTHER) and do not always include salience; the local summarizer guarantees offline execution but is extractive. The design trades recall/nuance for simplicity and reliability in a short assignment window.

With ADC configured, entities and sentiment populate fully. With only a Gemini API key, summaries still work and the system provides a useful agent answer with supporting snippets.

High-level Architecture

Figure: Data flows from CSV (or GCS) through preprocessing, GCP Language tools, and Gemini summarization. The agent composes retrieval → tools → synthesis.

Agent Flow Diagram

Could not render SVG agent flow; ensure svglib is installed.

Implementation Notes

- Configuration via .env; uses GOOGLE_API_KEY for Gemini path.
- Vertex path initialized with project/region; local summarizer guarantees no-internet runs.
- VS Code launches run module mode to keep package imports stable.
- Errors are surfaced in results CSV fields without stopping the run.

Extensions

Replace keyword retrieval with embeddings + cosine similarity; filter entity types and emit JSON; add more evaluation; and deploy as a Cloud Run service with GCS triggers.