# Intelligent Data Extraction & Summarization with Agentic Workflows on GCP

Developed by: Morteza Nagahi

## 1. Technical Approach & GCP Services Used

### Objective

The goal of this project was to design and implement a prototype system on **Google Cloud Platform (GCP)** that: - Extracts key entities and information from unstructured text. - Generates concise, human-readable summaries. - Demonstrates an **agentic workflow** where modular tools can be orchestrated to solve complex queries.

### Chosen Dataset

For prototyping, I selected a **news and customer review dataset** to simulate real-world unstructured text. Data was stored in **Google Cloud Storage (GCS)** and structured metadata was optionally ingested into **BigQuery**.

### GCP Services Used

* **Cloud Storage** – Raw text ingestion and storage.
* **BigQuery** – Structured queryable storage for document metadata.
* **Cloud Natural Language API** – Entity extraction, sentiment analysis, and key information retrieval.
* **Vertex AI Generative AI (Gemini)** – Abstractive summarization of documents.
* **Vertex AI SDK** – Unified Python interface for orchestrating NLP and generative tasks.

### Data Preparation & Exploration

* Performed cleaning (stopwords, punctuation, lowercasing).
* Exploratory Data Analysis (EDA) to understand document length, frequent entities, and sentiment distribution.

### Information Extraction & Summarization

* **Entity Extraction**: Cloud NLP API identified people, organizations, locations, dates, metrics, and sentiment.
* **Summarization**: Vertex AI Generative Models (Gemini/text-bison) generated fluent, human-like summaries.
* **Baseline Comparison (optional)**: Implemented extractive summarization using **TextRank** via SpaCy + NetworkX for benchmarking.

### Evaluation

* **Qualitative**: Summaries were coherent and captured key themes.
* **Quantitative**: (If dataset provides references) ROUGE scores used for evaluation.

## 2. Agentic Workflow Design

### Scenario: *Customer Insight Agent*

The agent helps businesses analyze product reviews to extract pain points, summarize customer sentiment, and generate actionable insights.

#### Agent’s Goal

Answer complex business queries like: - *“What are the top three customer complaints about product X in Q2?”* - *“Summarize positive customer feedback trends in Europe last month.”*

#### Tools Used

* **Document Retrieval**: BigQuery queries or Cloud Storage lookup.
* **Entity & Sentiment Extraction**: Cloud NLP API.
* **Summarization**: Vertex AI Generative AI.
* **Memory**: Firestore or BigQuery for storing historical context.

#### Reasoning & Planning Workflow

1. Accept a user query.
2. Retrieve relevant documents from BigQuery/GCS.
3. Extract entities, sentiment, and issues via Cloud NLP.
4. Generate summaries with Vertex AI.
5. Return insights in structured + natural language format.

#### Architecture Diagram (High-Level)

User Query → Agent API → BigQuery/Cloud Storage → Cloud NLP API → Vertex AI Generative Model → Structured Insights + Summaries

## 3. Productionization Approach

To extend the prototype into a production-ready solution:

### Scalability & Orchestration

* **Vertex AI Pipelines** for workflow automation.
* **Cloud Functions / Cloud Run** for serverless triggers.
* **Pub/Sub** for event-driven ingestion and processing.

### Security & Privacy

* IAM roles and least-privilege access for data.
* VPC Service Controls for isolation.
* Data encryption at rest (Cloud KMS) and in transit (TLS).

### Monitoring & Logging

* **Cloud Logging & Monitoring** for real-time observability.
* **Error Reporting** for system-level failures.
* **Custom Dashboards** for summarization/extraction performance metrics.

### Cost Management

* Prefer pre-trained APIs (Cloud NLP, Vertex AI) over custom training.
* Use **BigQuery slots & reservation models** for predictable pricing.
* Implement **batching & caching** to reduce repeated API calls.

### CI/CD & Reproducibility

* **Cloud Build + GitHub Actions** for CI/CD pipelines.
* **Terraform/Deployment Manager** for Infrastructure as Code.
* Version-controlled datasets and models.

## 4. Challenges & Trade-offs

* **Trade-off between abstractive vs. extractive summarization**: Abstractive models (Gemini) produce more human-like summaries but incur higher latency and cost compared to extractive baselines.
* **Quality evaluation**: Without gold-standard summaries, evaluation is primarily qualitative.
* **Scalability considerations**: While APIs are easy to use, high-volume workloads may require batch orchestration and caching strategies.

## 5. Conclusion

This project demonstrated: - End-to-end pipeline for text preprocessing, entity extraction, and summarization using GCP-native services. - An agentic workflow capable of handling multi-step reasoning across multiple documents. - A clear pathway to productionization with considerations for scalability, monitoring, cost optimization, and security.

The resulting system can serve as the foundation for real-world intelligent assistants such as **Customer Insight Agents**, **Research Assistants**, or **Support Ticket Analyzers**.