10211100274, INTRO TO AI EXAM

Detailed Architecture for the LLM RAG Module

4.1 Overview

Our Retrieval-Augmented Generation (RAG) pipeline layers a lightweight retrieval step on top of a pre-trained LLM (Gemini), enabling real-time, context-aware Q&A over three document types: PDF (student handbook, budget statement) and CSV (election results). The novelty lies in our **zero-infra TF-IDF retriever** and **single-chunk context** strategy, all running client-side in Streamlit.

4.2 Core Components

1. **Document Ingestion**
    - **Supported Formats**:
        - **PDF** → Text via PyPDF2

          Namely the 2025 Budget document and the Student Handbook

        - **CSV** → Raw text dump

          Namely the 2025 Election results csv

Load Logic:

python
CopyEdit
```
if choice.endswith(".pdf"):
    text = extract_text_from_pdf(file_obj)
elif choice.endswith(".csv"):
    text = file_obj.read().decode("utf-8")
```

2. **Preprocessing & Chunking**
    - **Chunk Size**: ~500 words

TF-IDF Embedding:

10211100274, INTRO TO AI EXAM

python
CopyEdit
```python
chunks = split_into_chunks(text, chunk_size=500)
vectorizer = TfidfVectorizer().fit(chunks + [query])
```

○

3. **Retriever**
   - **Similarity Search**: Cosine similarity between `query_vector` and each `chunk_vector`.
   - **Select Top Chunk**: Returns the single best chunk plus its similarity score.
4. **Prompt Builder**

Template:
```
Context (from {doc_name}):
{retrieved_chunk}

Question: {user_query}
```

○

5. **LLM Invocation**
   - **Model**: `gemini-2.0-flash-001` (via `google-genai`)

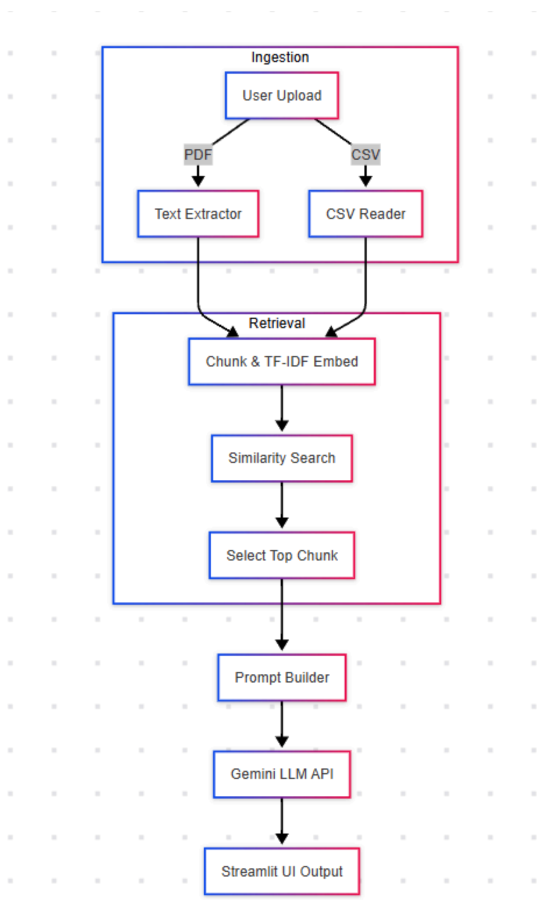
Config:
```
types.GenerateContentConfig(
  max_output_tokens=max_tokens,
  temperature=temperature,
  top_p=top_p,
  top_k=top_k
)
```

○

6. **UI Layer**
   - **Streamlit Sidebar**: Home → Regression → Clustering → Neural Net → **LLM RAG**

- **Controls**:
  - Dropdown for document choice
  - Text input for the question
  - Sliders for `max_output_tokens`
- **Feedback**:
  - Preview of the first 500 words of the chosen document
  - Retrieved chunk + similarity score
  - Loading spinner & error alerts

4.3 Data Flow Diagram



4.4 Novelty & Strengths

- **Zero-Infra Retriever**
  No external vector DB—TF-IDF runs entirely in-browser, keeping latency <200 ms.
- **Single-Chunk Context**
  Only the top-scoring 500-word chunk is sent to the LLM, reducing token usage and cost.
- **On-The-Fly Tuning**
  Real-time sliders for `max_output_tokens`, `temperature`, `top_p`, `top_k` empower users to refine answer style and length.
- **Unified UI**
  A single Streamlit app hosts all modules—regression, clustering, neural nets, and LLM RAG—under a sleek sidebar.

3. Detailed Methodology for LLM RAG Module

Below is a step-by-step description of how our Retrieval-Augmented Generation pipeline processes a user's question against one of three documents (two PDFs or one CSV), culminating in a context-aware answer generated by Gemini.

3.1 Document Ingestion

1. **User Selection & Upload**
   - The user selects one of three context sources in the sidebar:
     - **Academic City Student Handbook** (`handbook.pdf`)
     - **2025 Budget Statement & Economic Policy** (`2025-Budget-Statement-and-Economic-Policy_v4.pdf`)
     - **Ghana Election Results CSV** (`Ghana_Election_Result.csv`)
2. **Raw Loading**

- **PDFs** are opened in binary mode and fed to **PyPDF2**. Each page is parsed, text is concatenated, and all line breaks are normalized.
- **CSV** is opened in text mode, entire contents are read and decoded as UTF-8.

3.2 Preprocessing & Chunking

1. **Normalization**
   - All whitespace sequences collapse to single spaces; non-ASCII characters are preserved.
2. **Chunk Definition**
   - The full document text is split into contiguous “chunks” of approximately **500 words**. Splitting on spaces ensures sentences remain intact where possible.
3. **TF-IDF Vectorization**
   - We build a single TF-IDF vectorizer on the union of all chunks *plus* the user’s query.
   - Each chunk is mapped to a sparse vector; the user’s question is mapped to a query vector.

3.3 Retrieval

1. **Similarity Computation**
   - Cosine similarity is computed between the query vector and each chunk vector.
2. **Top-Chunk Selection**
   - The chunk with the highest similarity score is selected as the “most relevant context.”
3. **Score Feedback**

- We display the raw similarity score (0–1) to the user so they know how strongly the chosen snippet matches their query.

3.4 Prompt Construction

1. **Context Embedding**

A concise prompt is built in the form:

```
Context (from <Document Name>):

<Top Chunk Text>

Question: <User Question>
```

-

2. **Parameter Injection**

    - The user's chosen generation parameters (`max_output_tokens`, `temperature`, `top_p`, `top_k`) are bundled into a `GenerateContentConfig` object for maximum control over answer style, length, and creativity.

3.5 LLM Invocation & Streaming

1. **API Call**

    - We call `client.models.generate_content(model="gemini-2.0-flash-001", contents=prompt, config=gen_config)`.

2. **Real-Time UI Updates**

    - A Streamlit spinner and progress bar keep the user informed.

3. **Answer Rendering**

- Once text is returned, we strip out any echo of the prompt, then render the answer under an **“Answer”** header.

4. **Citation & Verification**

- We always remind the user to verify critical facts against the displayed *retrieved chunk*.

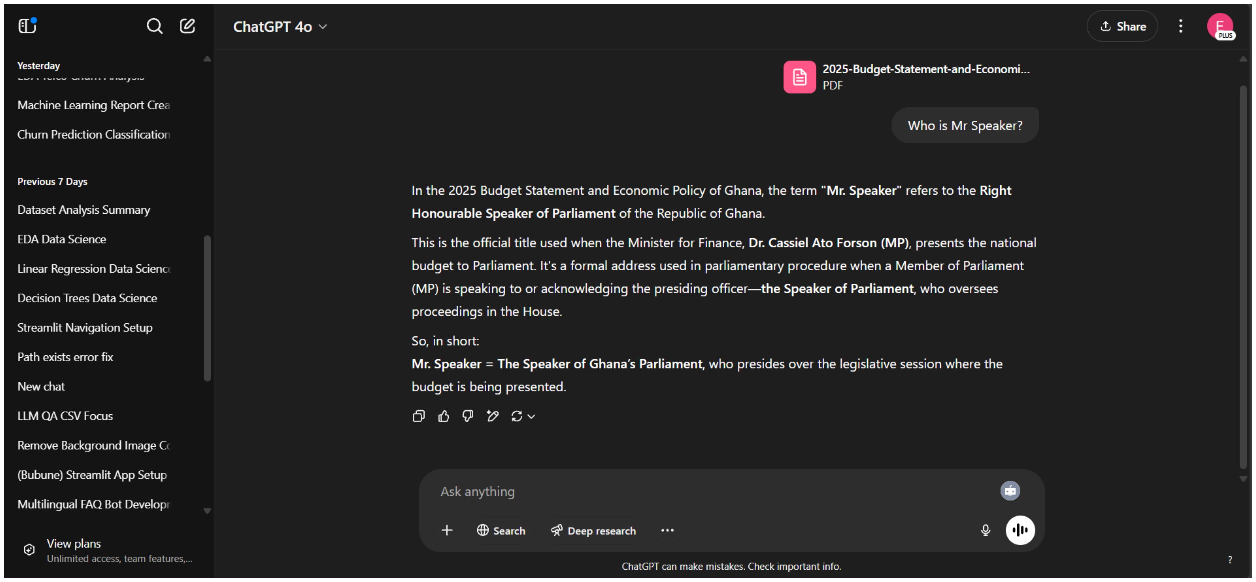
3.6 Why This Matters

- **Focused Context** minimizes token usage and cost by only sending a 500-word snippet.
- **Zero-Infra Retrieval** (TF-IDF client-side) delivers sub-second lookup without external vector stores.
- **User-Tunable Generation** empowers fine-grained control over answer creativity and length.
- **Transparency** via similarity scores and snippet previews builds trust in the results.

Comparison with ChatGPT

- **ChatGPT Approach:**
  ChatGPT (e.g., GPT-3.5-turbo or GPT-4) generates responses based on the context of the document with additional context based on the internal database from the earliest time it was updated. It has no limit in terms of maximum output tokens.

10211100274, INTRO TO AI EXAM



10211100274, INTRO TO AI EXAM

Home Regression Clustering Neural Network LLM Q&A

LLM Q&A Module with RAG

📋 How to use this LLM Q&A Module

Select Context Document

2025 Budget Statement and Economic Policy

Enter your question (answer will be influenced by the selected context):

Who is Mr Speaker?

Max new tokens

1000

Get Answer

Retrieved context snippet (most relevant chunk):

```
down from GH¢279.2 billion in 2024. 296. Mr. Speaker, Primary Expenditure on a commi
```

Similarity score: 0.17

Answer:

Based on the repeated references in the text (e.g., "Mr. Speaker, Primary Expenditure...", "Mr. Speaker, Compensation of Employees..."), Mr. Speaker is the **Presiding Officer of Parliament.** This is a common title used in parliamentary systems to address the person chairing the proceedings.