

Airline Reviews RAG Project

Key Architectural Decisions:

I implemented a primary retrieval flow in langflow, while focusing on ingestion via file upload in AstraDB in order to test out vector search, plus have control over similarity (dot-product/cosine). However given the requirements listed, I did design an ingestion workflow that can optionally be tested out.

Design choice for Retrieval:

Input Node:

Text-based input for natural language queries. This is where the user inputs their questions.

Astra DB Vector Store:

This is the node that lets the tester add their AstraDB database token and specify the database and collection where the vectorized data has been ingested.

Parse Data Node:

This node just makes sure that the data parsed from the documents is in text form before it is passed to the prompt node.

Prompt Node:

This node has a default context and question format, where the connection of the chat input is specified as my question variable, and the parsed documents from AstraDB are the context.

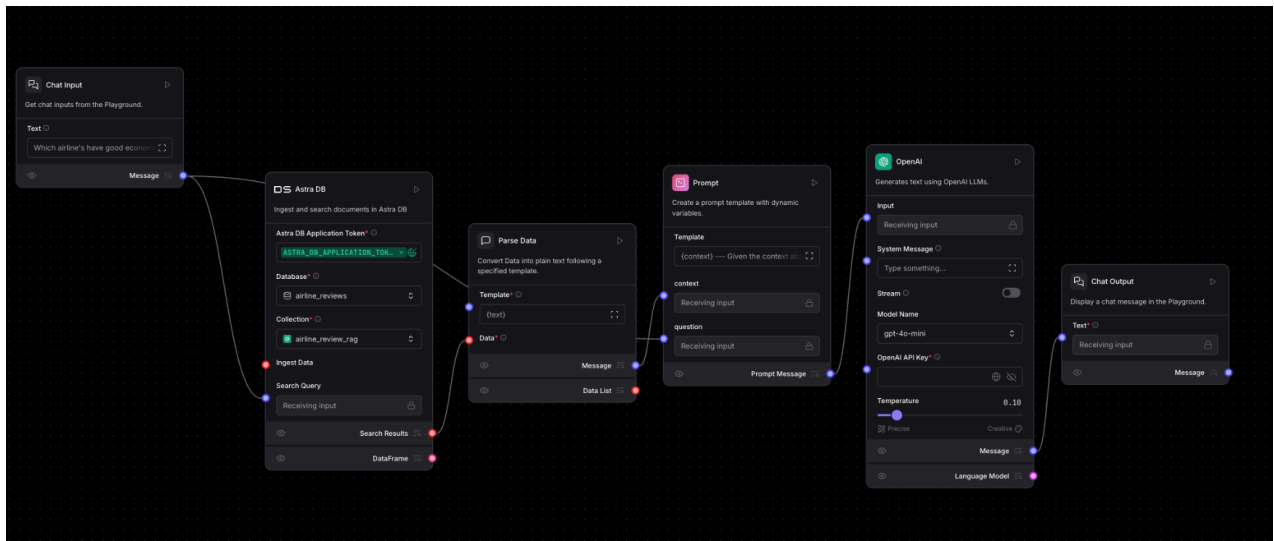
LLM Node (OpenAI):

I decided to use an open ai client for the LLM based NLP task, and so this LLM node requires an OpenAI API key. Given a generated prompt from the prompt node, this will use the gpt-4o-mini model to analyze the reviews and the question and answer it in a RAG oriented way.

Output Node:

The output message generated by the LLM node will be displayed here.

Final Retrieval Flow Architecture Screenshot:



Design choice for Ingestion:

File Node:

File-based input node to allow user to import a csv or pdf file. In our case we needed to import text-based unstructured data.

Split Text Node:

This node implements chunk-sized splitting to make long inputs into retrievable, embed-friendly blocks. This greatly reduced the strain it would take to embed the entire document.

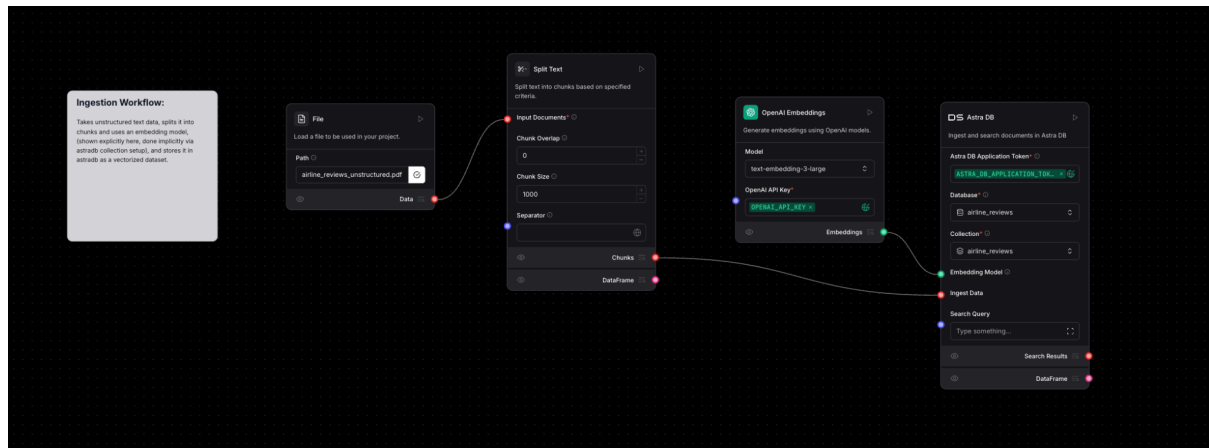
Embedding Node:

This node implements embedding using an OpenAI client. This also requires an OpenAI API key, and for my case I used the text-embedding-3-large model.

AstraDB Vector Store Node:

This is the node that lets the tester add their AstraDB database token and specify the database and collection where the vectorized data has been ingested.

Final Ingestion Flow Architecture Screenshot:



Challenges & Solutions:

Initially, while setting up ingestion via a Python script, I had issues connecting to the AstraDB database. I was able to fix this by installing the right dependencies.

Next after successfully ingesting the data, I had a problem with embedding it as I was performing the embedding within my Python script using OpenAI's API. The script would run for about 20 minutes and then say it failed to update embeddings. This would prove to be a fatal issue, which made me pivot to the other option of directly ingesting and vectorizing the data using Langflow/AstraDB.

I decided to upload the data directly to AstraDB as I saw how could control the similarity scoring and perform vector search to see how the dataset is vectorized.

Lastly, I successfully created a retrieval flow pointing to my existing database and collection; however, when I added an ingestion flow in case it would be part of our requirements, the API credentials kept vanishing from the project, leading to the flow not working. This was especially curious that my retrieval flow worked perfectly fine with the same credentials. After multiple tries to refresh and retry, I decided to just add the ingestion flow to the project to demonstrate that I understand how it works, and perhaps if tested by another reviewer, it will work successfully with their credentials.

Screenshots of results:

Retrieval queries in Playground:

**User**

Which airlines are delayed most often according to reviews?

**AI** gpt-4o-mini

According to the reviews, the airlines that are delayed most often are:

1. **Philippine Airlines** - Noted for domestic flights always being delayed.
2. **Jetstar Pacific** - Mentioned that all three booked flights were rescheduled multiple times and had significant delays.
3. **Southwest Airlines** - Consistently delayed flights over a period of seven months, causing additional costs and stress.
4. **SpiceJet** - Described as having poor time management with the last five flights resulting in delays.

These airlines received negative feedback specifically regarding their punctuality and reliability.

**User**

which airline has good reviews for flights from New York to Los Angeles?

**AI** gpt-4o-mini

The airline that has good reviews for flights from New York to Los Angeles is United Airlines. One review highlights the flight experience on United, mentioning the new 787-10 aircraft as the best domestic transcontinental equipment available, along with on-time departure and arrival. However, it should be noted that the review also mentions mediocre ground service at EWR, which may not reflect the overall experience.

Future Improvements:

In the future I could:

- Add more data for a more comprehensive RAG system.
- Add metadata filters (e.g., by airline or route)
- Make it compatible with multi-modal data where the data can include pictures of flight seats, food service, amenities, etc.
- Create a frontend UI to integrate the flow using Langflow API.