Context-Aware Chatbot Using n8n, Pinecone and Google Drive

In today's AI-driven world, the ability to retrieve **accurate, context-rich answers** from large collections of documents is a game-changer. This blog details the creation of an **Agent-powered RAG (Retrieval-Augmented Generation)** workflow within n8n, designed to extract specific details from Google Drive PDFs and provide precise, context-aware responses without manual searching, thereby automating and simplifying repetitive, time-consuming, and error-prone data processing tasks.

Problem Statement

Searching through multiple PDFs manually is inefficient. The problems include:

- Slow retrieval due to manual file opening and keyword searching.
- Missed context when using traditional search tools.
- Inability to scale when document counts increase.

The challenge was to build a system that:

1. Ingests both existing and new PDFs from Google Drive.
2. Extracts their content and stores it in a vector database.
3. Responds to user queries with accurate, context-aware answers.

Approach & Methodology

The approach uses **RAG (Retrieval-Augmented Generation)** principles:

- **Retrieval**: Search for relevant content in a vector database.
- **Augmentation**: Feed the retrieved context to an AI model.
- **Generation**: Produce natural language answers.

The methodology involves two interconnected workflows:

1. **Data Ingestion** —Ingest PDFs, extract content, split text, generate embeddings, and store them in a vector database
    a. **Bulk ingestion** for all existing files.
    b. **Incremental ingestion** for new files detected in real time.
2. **Query Processing** — Accept user queries, search embeddings for relevant content, and generate responses.

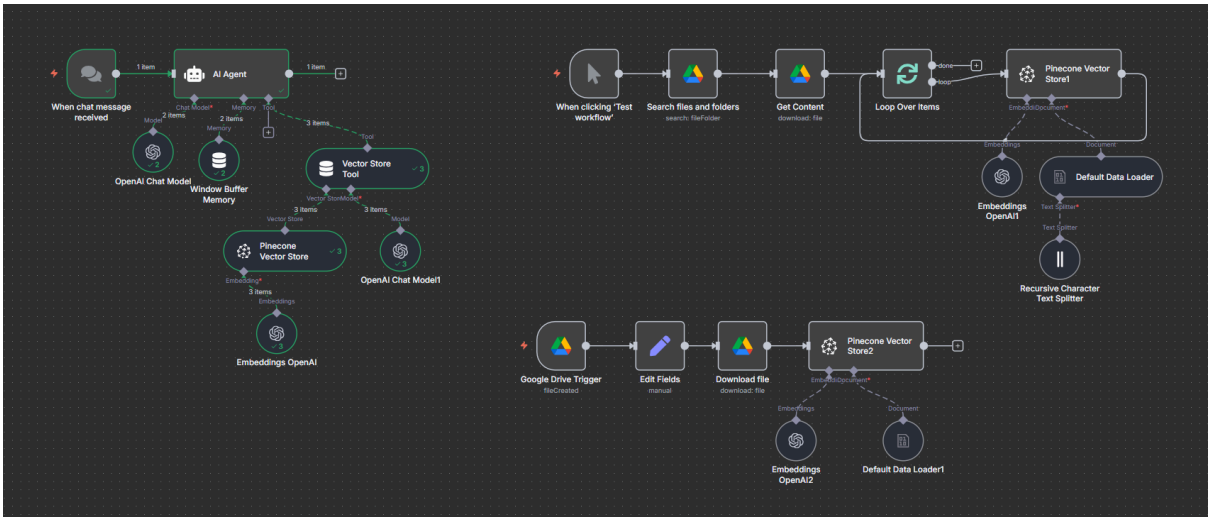
Tools, Frameworks, and Data Used

- **n8n** – Automation workflow platform.
- **Google Drive Trigger & Search Nodes** – To detect and fetch files.
- **PDF Parser** – For extracting text from PDFs.
- **Recursive Character Text Splitter** – For splitting content into manageable chunks.
- **OpenAI Embeddings API** – For creating vector embeddings.
- **Pinecone Vector Store** – For storing and retrieving vectors.
- **OpenAI Chat Model (GPT)** – For generating responses using retrieved context.
- **Data Source** – PDFs stored in Google Drive.

Workflow Overview

The goal was to create an **intelligent chatbot** that:

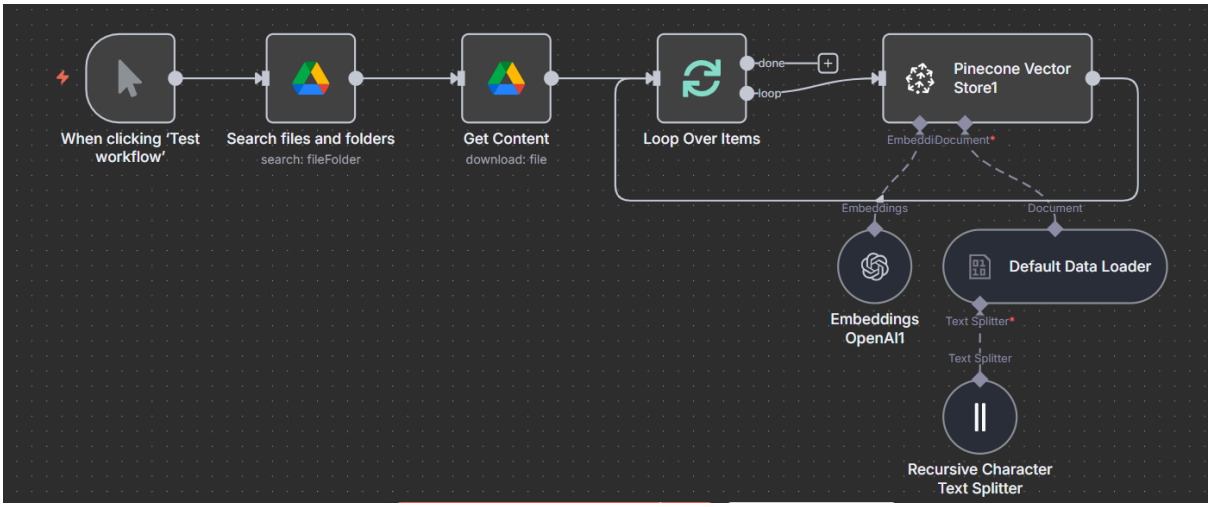
1. Monitors **Google Drive** for new or updated PDFs.
2. **Extracts and processes** the text content from those PDFs.
3. Converts the text into **vector embeddings** for efficient semantic search.
4. Stores the embeddings in a **RAG vector database**.
5. Uses an AI Agent to retrieve relevant context and answer user questions.



Implementation Steps

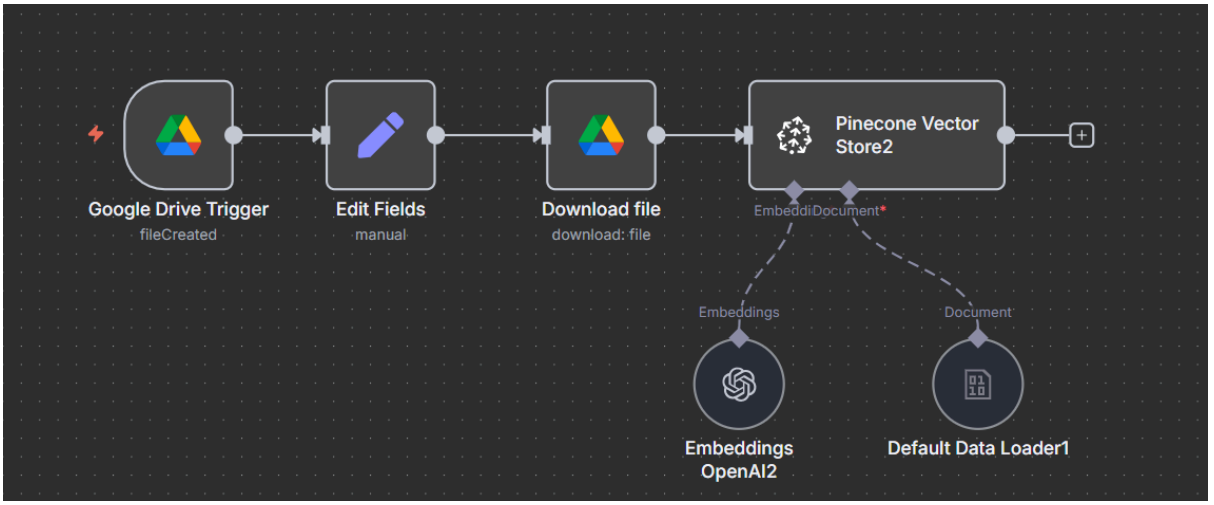
Workflow A – Bulk Ingestion (All Existing Files)

1. **Trigger** – Manual or scheduled execution.
2. **Search Files & Folders** – Retrieves all PDFs from Google Drive.
3. **Download Content** – Fetches each file's content.
4. **Text Splitting** – Breaks the document into chunks using Recursive Character Text Splitter.
5. **Embedding Generation** – Uses OpenAI to create embeddings.
6. **Vector Storage** – Saves vectors and metadata in Pinecone.



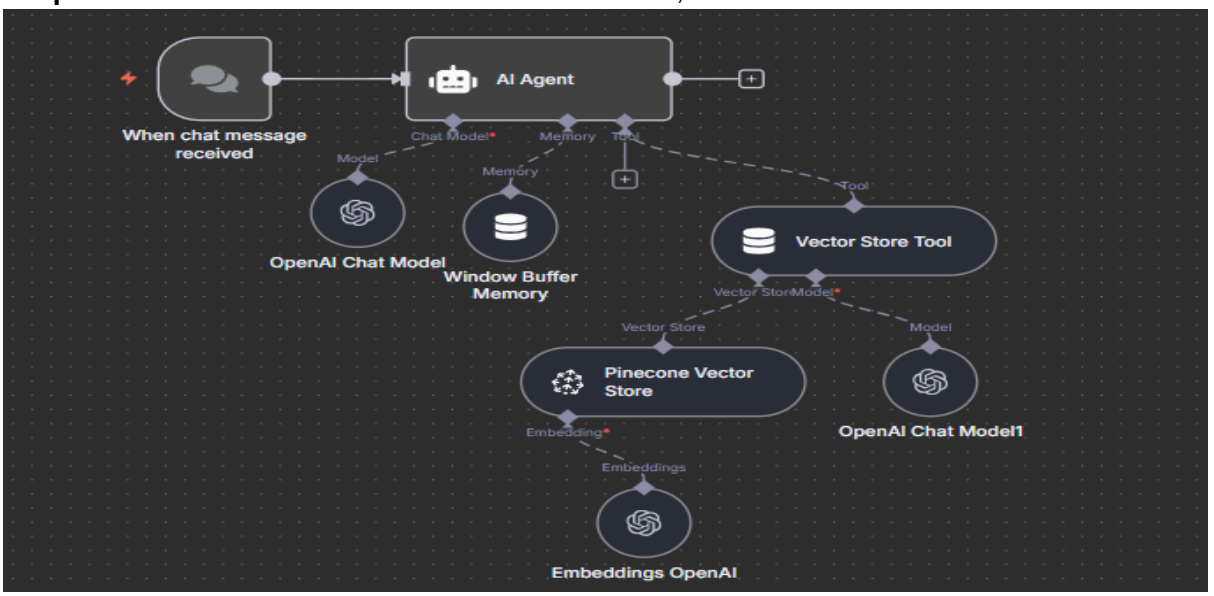
Workflow B – Incremental Ingestion (New Files Only)

1. **Trigger** – Google Drive fileCreated event.
2. **Download File** – Retrieves the new file content.
3. **Text Splitting** – Processes content into chunks.
4. **Embedding Generation** – Creates embeddings for the chunks.
5. **Vector Storage** – Adds embeddings and metadata to Pinecone without reprocessing existing files.



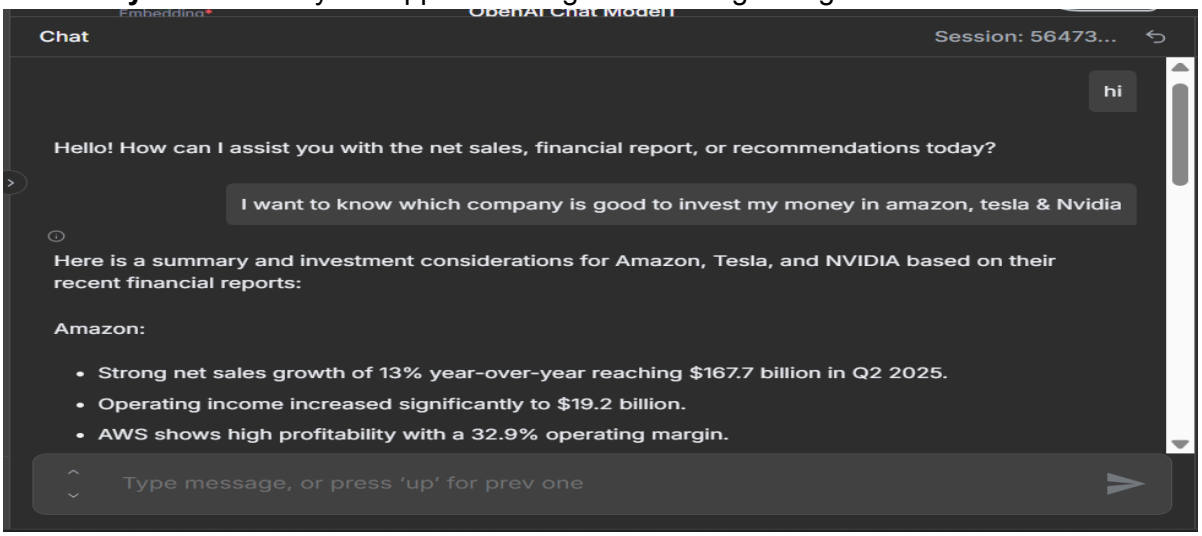
Query Workflow

1. **Chat Message Received** – User sends a query.
2. **AI Agent** – Processes the query and decides retrieval strategy.
3. **Vector Store Tool** – Searches Pinecone for relevant chunks.
4. **Context Injection** – Feeds results into the AI chat model.
5. **Response** – Returns an accurate, context-aware answer.



Results and Outcomes

- **Automated Processing** – Both old and new files are handled without manual effort.
- **Fast Query Responses** – Relevant answers are delivered in seconds.
- **Improved Accuracy** – Context-aware retrieval ensures better answers.
- **Scalability** – Easily supports large and growing document sets



Future Improvements / Next Steps

- Support more file types (Word, Excel) and database connections.
- Add triggers for file updates to keep content fresh.
- Integrate chatbot into a web app for easier access.
- Refine retrieval accuracy and embedding performance.

Conclusion

The n8n RAG Workflow Chatbot transforms stored documents into an always-updated, searchable knowledge base. By automating ingestion and delivering context-rich answers in seconds, it saves time, boosts accuracy, and makes information instantly accessible across teams.

References

- OpenAI API Documentation: https://platform.openai.com/docs
- Pinecone Documentation: https://docs.pinecone.io
- n8n Documentation: https://docs.n8n.io