GenAI Accelerators to improve cost savings and CSAT in Airline Operations

NUS National University of Singapore
School of Computing

Author: Sai Niharika Naidu Gandham, M.S Computing
niharikanaidu.gandham@gmail.com

Connect with me

E-Validator

(OCR & GenAI based automation saving >30k SGD per month)

Introduction

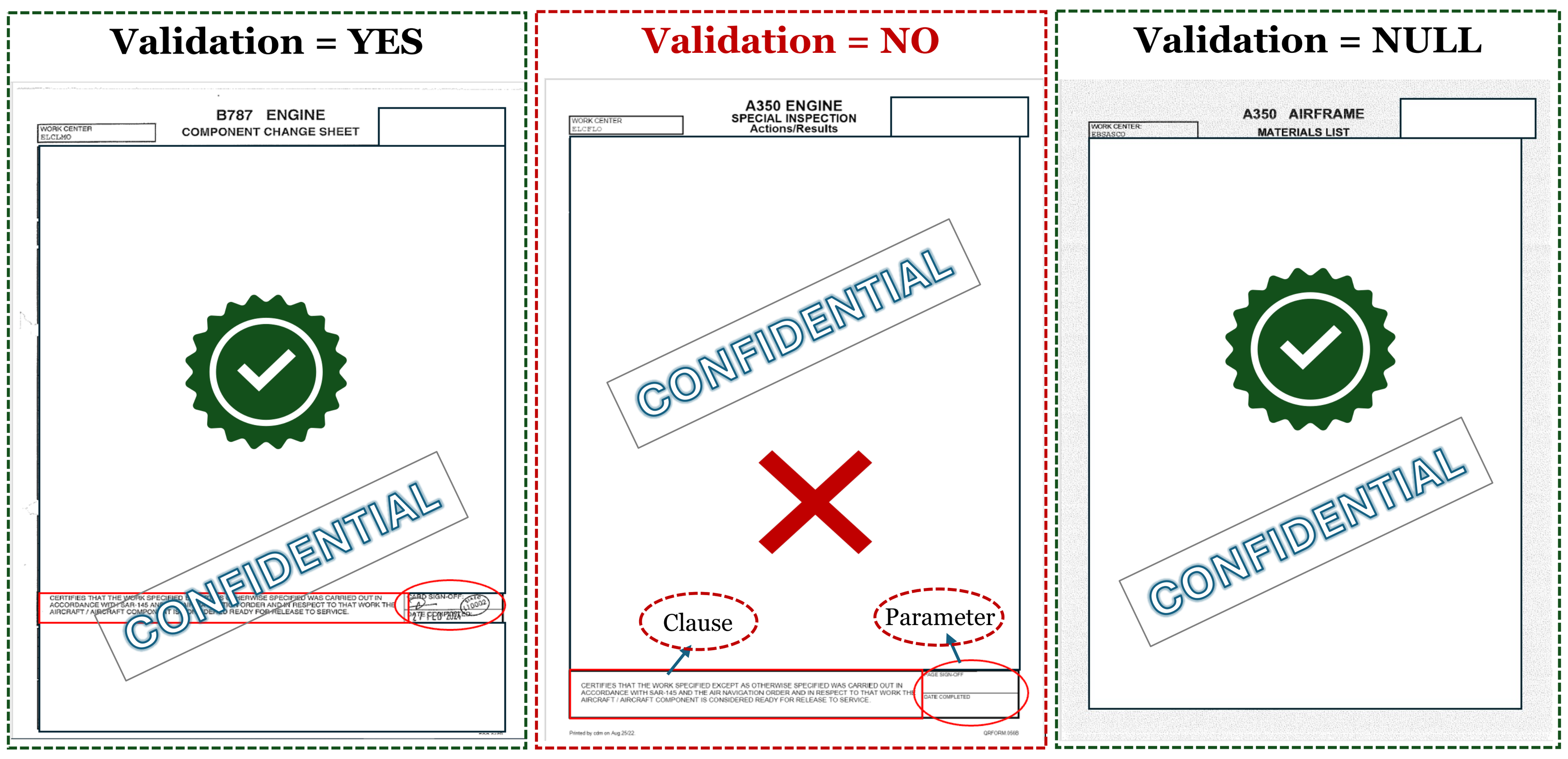
'**E-Validator**' is designed as an essential tool for bay planners to streamline the Taskcard validation process that is a manual and time consuming

Challenge	Solution
Manual and time consuming (dedicated resources)	Automated using GenAI (>30k SGD savings per month)
High volume of Taskcards ~500 (52 hours per day)	Only 7 seconds per Taskcard (~40x efficient)
Prone to Human Error	Stable and accurate solution (100% accuracy)

Methodology

Initial Results

Below are sample task cards and the highlighted sections are ***clause and parameter*** sections. Three crucial checks are done on the parameter section for **Handwritten signature, Stamp and Date** at the CARD SIGN-OFF and DATE COMPLETION.



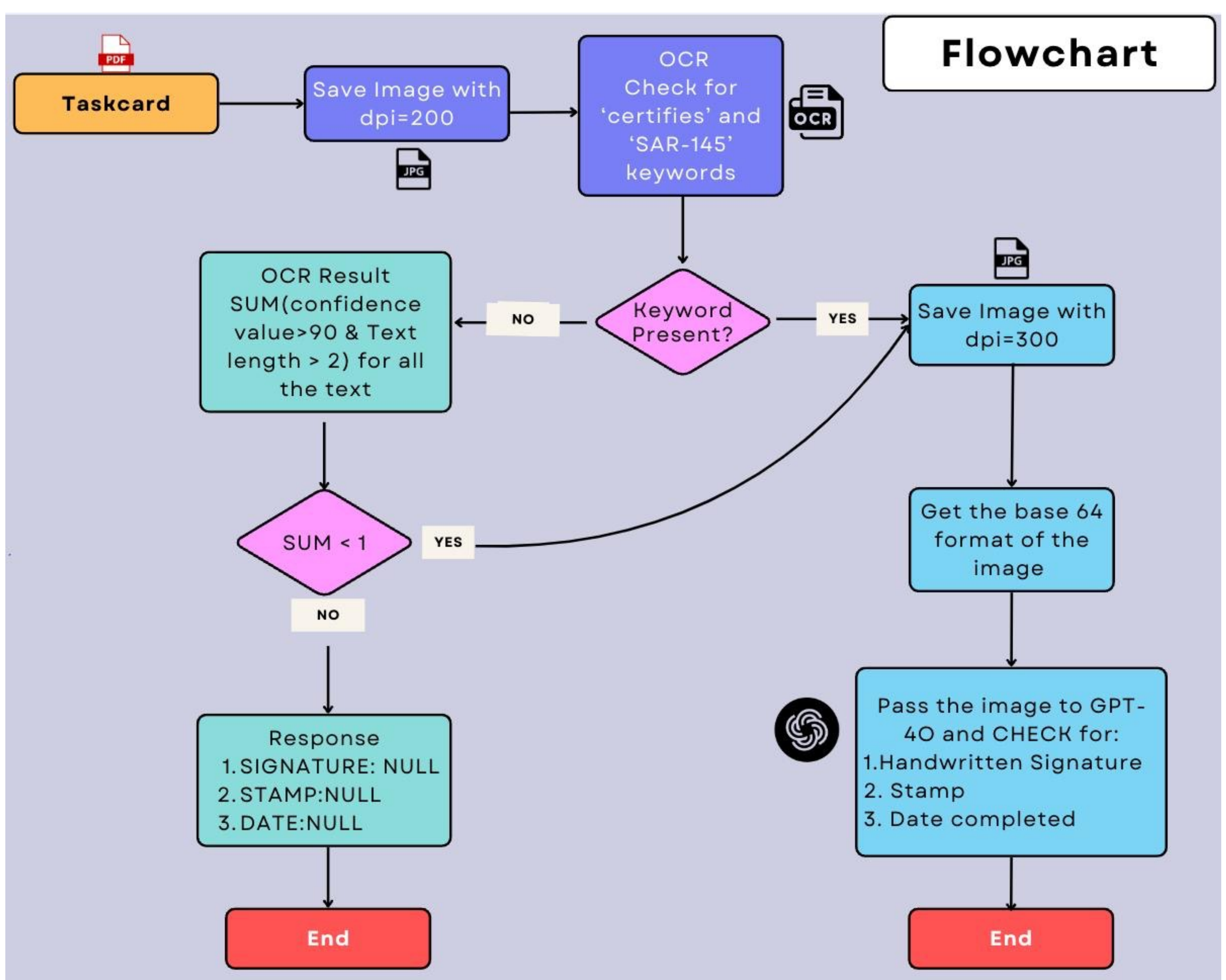
Model	Accuracy
Haiku	50.2%
Sonnet	70.8%
GPT-4O	91.1%

When images are tested on the models:

- **Haiku** is least performing model with highest number of False positives
- **Sonnet** has relatively smaller number of False positives compared to Haiku
- **GPT-4O** is the best performing model with zero False positives, accuracy is low due to False Negatives
- Experiments are conducted to find the coordinates of Parameter section, crop the image and send to the model. However, the position of this section is fluid across task cards.

Solution

- ✓ In the initial results, images sent to GPT-4O resulted in false negatives (FN) due to 'NO' responses in NULL scenarios.
- ✓ To counter this, **OCR was introduced to filter NULL cases**. OCR checks for keywords in the 'clause'; as clause is always present on a Taskcard with parameter section.
- ✓ This ensures OCR will filter out NULL cases to a different branch bypassing GPT-4O check
- ✓ This allows GPT-4O to respond only with YES/NO improving the accuracy



Conclusion

- With the introduction of OCR, our solution now boasts 100% accuracy if images are clear and signatures, stamps, and dates are visible
- The groundbreaking solution developed as a proof of concept, has already saved the client ~**220 man-hours** in the first review after deployment
- As we prepare for deployment as full-fledged solution, we anticipate a **70-80% reduction in man-hours resulting in >30k SGD savings per month**. This innovation is set to revolutionize efficiency and productivity, showcasing the transformative power of technology.

Engineer Assist

(RAG based search & recommendation engine improve CSAT)

Introduction

'**Engineer Assist**' is a search and recommendation engine leveraging a library of manuals to aid on-field engineers in aircraft repair and maintenance. It serves as a personal assistant, providing targeted recommendations for specific issues.

Challenge	Solution
Siloed large libraries of manuals (30+ large manuals)	Connected and comprehensive recommendation solutions
Manual and delayed turn around time	Assisted and improved turn around time (CSAT)
Human dependent and siloed with no evolution	Learning model that also leverages on historic/log data

Methodology

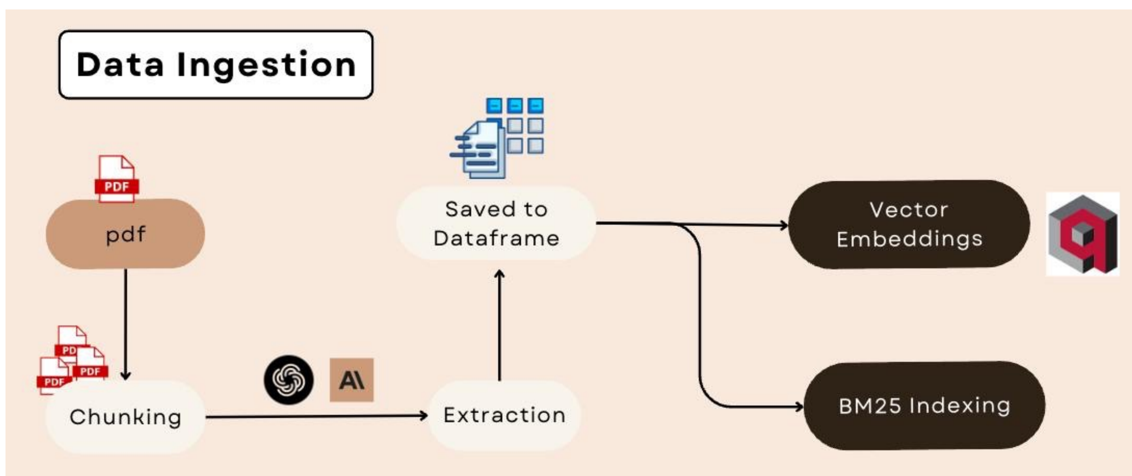
Initial Setup

Data Ingestion in RAG System : *"Retrieval-Augmented Generation (RAG) systems rely heavily on effective data ingestion to enhance knowledge retrieval. This ensures all data is accurately captured and processed."*

- ✓ **Multi-PDF Input**: Data often resides in multiple large PDF files, each over 1,000 pages.
- ✓ **Chunking**: PDFs are split into manageable chunks based on chapter or sections. Chunks are designed so that no text is spanned between different chunks.
- ✓ **Data Extraction**: Using AI models like **Claude** and **GPT**, data is extracted from each chunk. When formatting is consistent, data is extracted based on section names. Otherwise, the extraction is done in **dictionary** or **Markdown** format.
- ✓ **Prompt Design**: Custom prompts are designed using <scratchpad> and <output> tags for each knowledge base to optimize data retrieval.

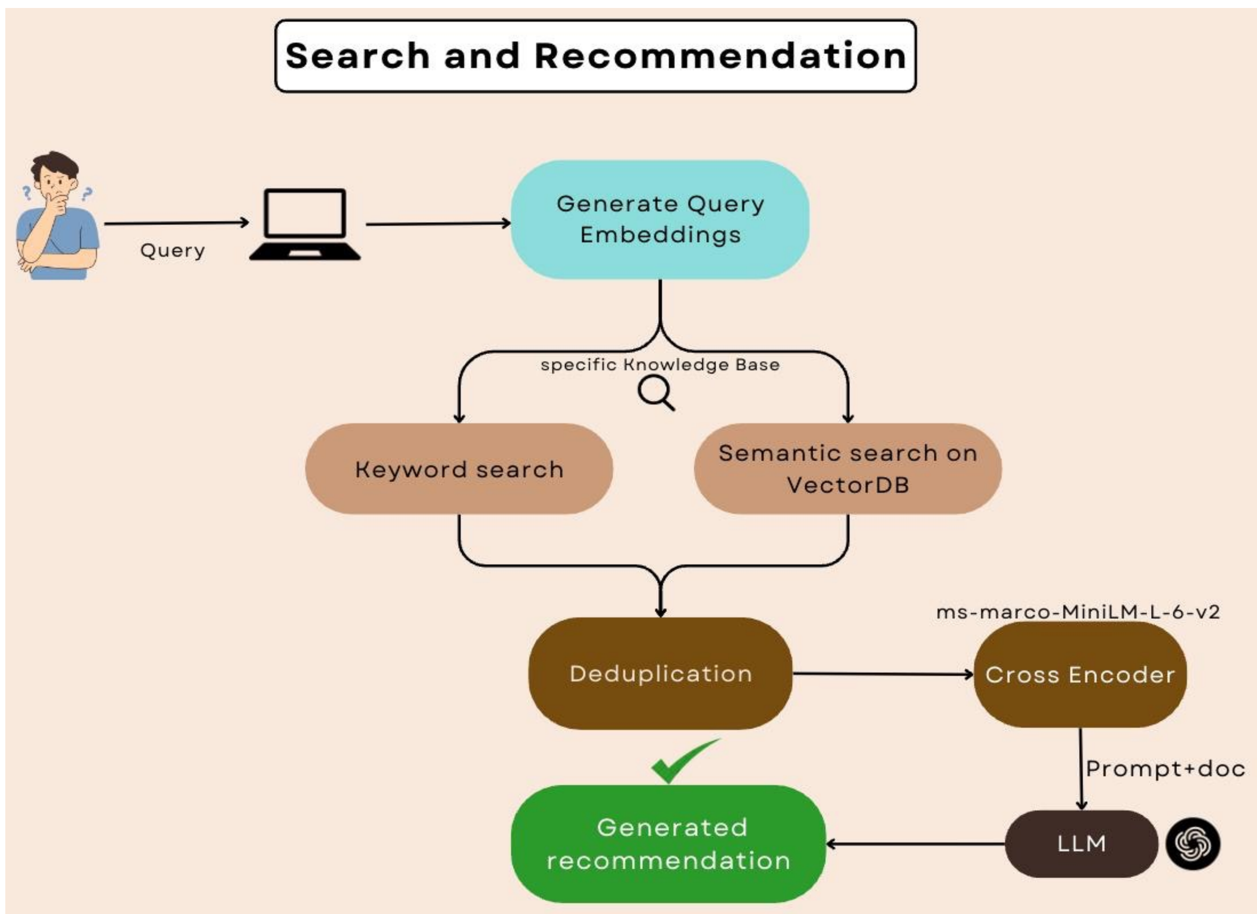
After data extraction is completed,

- ✓ **Vector Embeddings**: Generated using OpenAI embeddings and saved into **Qdrant** for efficient vector-based retrieval.
- ✓ **Keyword Search**: **BM25 indexing** is applied using Rank BM25, and further **OpenSearch** can be used to enhance search performance



Search & Recommendation

a. Embeddings are generated for the user query to perform **semantic search** in the **Qdrant** Database. Query is tokenized to perform **Keyword search** on **BM25** indexed data.
b. Deduplicate retrieved documents and use a **cross-encoder** to generate relevance scores for query-document pairs and rerank them.
c. Pass the most relevant documents to the **LLM (Large Language Model) to generate a response**.



Conclusion

- Recommendations are generated separately for each Knowledge Base by the LLM
- By leveraging on cutting edge technology, the support team's efficiency is improved which further enhances overall task management
- The evolution recommendation model can include data of multiple aircraft models to become a one stop solution/assistant for all field engineers. This can hugely improve the resource allocation across aircraft models.