Project Report: NUST Bank Chatbot (RAG-Powered)

This report details the development of a Retrieval Augmented Generation (RAG) powered chatbot for a fictional entity, NUST Bank. The project aims to enhance customer service by providing an AI-driven assistant capable of handling customer inquiries based on a curated knowledge base.

Project Overview

The NUST Bank Chatbot is designed to be a responsive AI assistant for a fictional bank. The core objective is to leverage a Large Language Model (LLM) and a RAG architecture to provide accurate and context-aware responses to customer queries, utilizing the bank's product knowledge. The project emphasizes maintaining data privacy and trust while converting anonymized customer interaction documents into a functional chatbot.

Completed Step: 2. LLM Implementation

We have successfully completed Step 2 of the project: LLM Implementation. This involved implementing the core Large Language Model component and developing a prototype system capable of returning initial, LLM-driven answers based on the bank's product knowledge. As part of this step, the system architecture has been defined and is represented in the accompanying Architecture Diagram.

The implementation and prototype system are detailed in the following sections.

Data Preparation and Extraction

The project utilizes a dataset from the fictional NUST Bank, provided in an Excel file named `NUST Bank-Product-Knowledge.xlsx`. To make this data usable for training an LLM, a data extraction process was implemented as part of Step 1 (Data Ingestion & Preprocessing, not explicitly detailed in the provided files but implied as a prerequisite).

The `QA_Extraction_Initial_finetuning_Data.ipynb` notebook outlines this process. It involves:

1. Reading the `NUST Bank-Product-Knowledge.xlsx` file using the pandas library.
2. Iterating through each sheet of the Excel file.
3. Converting the content of each sheet into a string format.
4. Using the Gemini-2.0-flash model via the Google GenAI API to extract structured Question and Answer pairs from the text. A specific JSON schema `{'question': str, 'answer': str}` is used to guide the extraction.
5. Collecting all extracted QA pairs into a single list.
6. Saving the compiled list of QA pairs into a JSON file named `qa_pairs.json`.

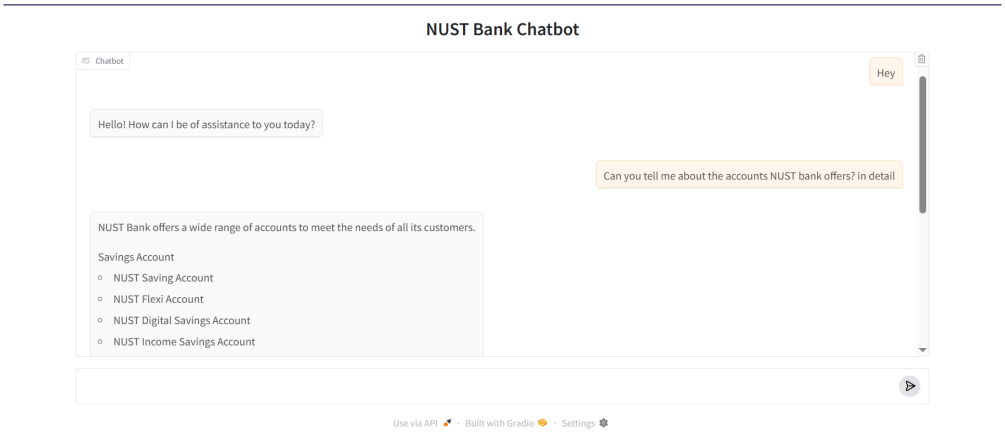
The qa_pairs.json file serves as the structured dataset for finetuning the language model, containing 414 question-answer pairs as demonstrated in the notebook output.

Model Finetuning

The central component of the chatbot's initial response generation is a finetuned Large Language Model. The Phi3_5_mini_Initial_Model_Finetuning.ipynb notebook details the finetuning process using the Unsloth library for efficiency, which was a key part of the LLM implementation.

Chatbot Interface

The notebook includes the code to build and launch this interface.



NUST Bank Chatbot

Hey

Hello! How can I be of assistance to you today?

Can you tell me about the accounts NUST bank offers? in detail

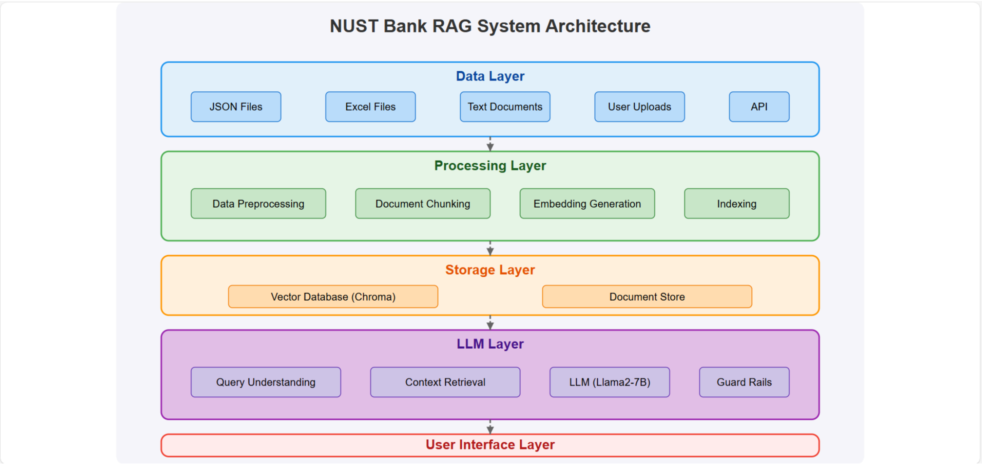
NUST Bank offers a wide range of accounts to meet the needs of all its customers.

Savings Account

- NUST Saving Account
- NUST Flexi Account
- NUST Digital Savings Account
- NUST Income Savings Account

Architecture Diagram

The architecture of the system, illustrating the flow of data from the product knowledge base through the data extraction, model finetuning, and inference stages to the user interface, has been defined and is submitted as part of this step.



Conclusion

With the completion of Step 2, we have successfully implemented the core LLM component by finetuning a Phi-3.5-mini model on extracted bank product knowledge data. A functional prototype system, accessible via a Gradio interface, has been developed to demonstrate the model's ability to provide initial, LLM-driven answers. The system architecture has also been documented. The project is now positioned to integrate the RAG capabilities in subsequent steps to enhance the chatbot's accuracy and contextuality further.