In20-S5-CS3501 Data Science and Engineering Project
Department of Computer Science and Engineering
University of Moratuwa

Localised Chatbot for Bank Customer Care

Bridging Language Barriers for Enhanced Banking Support

Team Members:

1. ANOSAN J. 200040B
2. AMPAVILA I.A. 200036T
3. AYESHA M.M.N. 200058L

Group ID: 1
Project ID: 8
Mentor: Dr. Buddhika Karunarathne

1. Executive Summary:

This project aims to enhance customer support services for Sri Lankan bank customers who communicate in native languages, by developing an intelligent trilingual chatbot as a virtual assistant. A diverse dataset of Sinhala, Tamil and English language interactions between bank customers and support agents will be used as data to train the chatbot. The dataset will include various banking queries, account-related questions and general inquiries. Natural Language Processing techniques will be used to preprocess and analyze the dataset. The chatbot will also use neural network models and transformer architectures to understand and generate responses in the native languages. The chatbot will serve as an interactive virtual assistant, providing quick and accurate responses to customer queries. It will offer information on banking services, account-related inquiries, and general questions, increasing customer satisfaction and reducing the load on human support agents. The project's success will be measured by improved customer engagement, reduced response times, and increased efficiency in handling customer support interactions in all three languages.

2. Problem Statement:

This project addresses the challenge of providing efficient customer support to users who prefer communicating in Sinhala or Tamil languages in Sri Lanka. The existing mostly English-centric customer service causes communication barriers and inefficiencies, leading a less satisfactory customer experience. There are many customers who tend to not understand bank services and opportunities to the fullest. The bank may lose potential customers due to this reason and an increase in customer inquiries can also occur.

The main objective of this project is to develop an intelligent chatbot that can, in a comfortable language, automate responses for customer questions and provide personalised assistance in simple tasks while maintaining data security and compliance with banking regulations at all times..

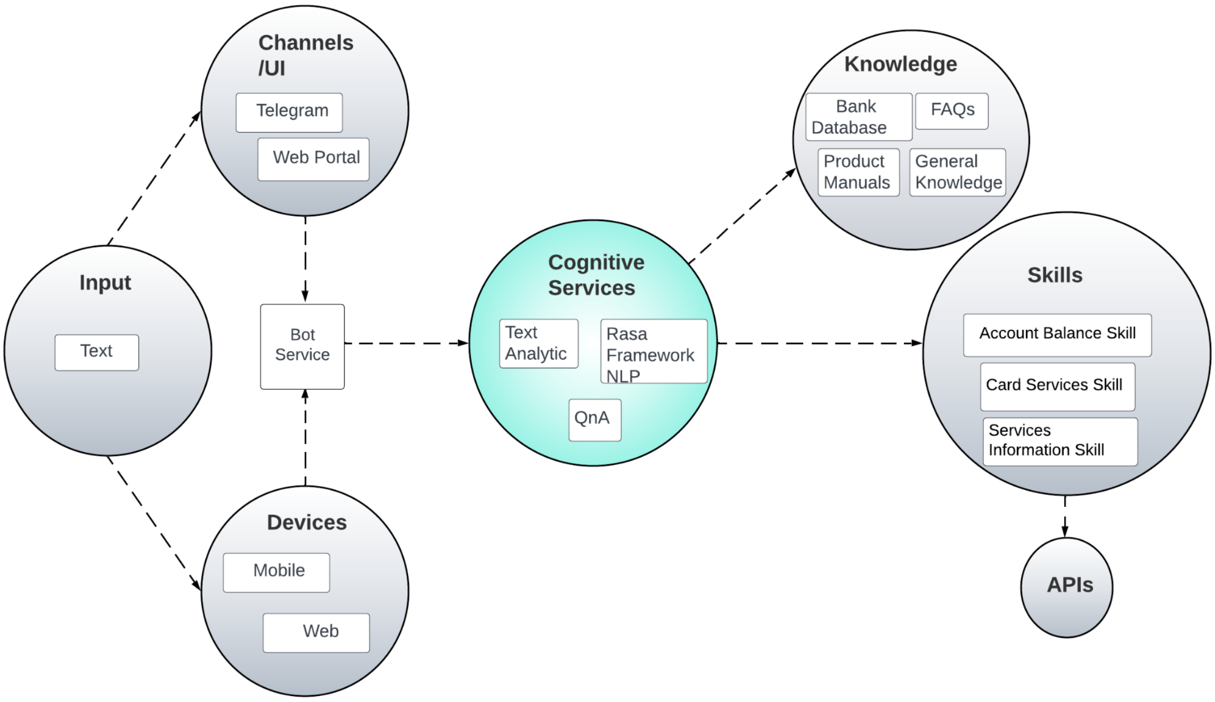
3. Data Description:

The data for this project can be collected from various sources such as Customer support logs, Customer feedback, FAQs and Knowledge Base, databases, web blogs, articles, YouTube video transcriptions, podcasts, tweets, LinkedIn posts, and files of different formats among others.
These collected data will include a mix of structured and unstructured data. Structured data may include timestamps, customer IDs, transaction IDs, etc. Unstructured data includes textual interactions between customers and support agents, as well as customer feedback.

The data can be collected from the bank's existing customer service interactions and stored in a structured format, including the user queries and corresponding responses. Frequently Asked Questions (FAQs) can be curated and annotated with their respective intents and entities to create a labelled dataset. Sample conversations in Sinhala or Tamil can be gathered from real users engaging with a prototype chatbot or by using data augmentation techniques on existing data. Banking-related information from the bank's website or documentation can be used to augment the knowledge base of the chatbot.

4. Methods:

The Localised Chatbot for Bank Customer Care project will employ a combination of data science methods, natural language processing (NLP) techniques, and software development practices to achieve its objectives. The components of the conversational chatbot are shown below.



The methods involved can be broadly categorized as follows:

Data Cleaning and Preprocessing: Prepare the collected data by performing tasks such as text normalization, tokenization, and removing noise. This step is crucial to ensure the quality and accuracy of the training dataset.

Natural Language Processing and Model Development:

NLP Framework Selection: Utilize the Rasa framework, a powerful open-source NLP framework, to develop the chatbot's language understanding and dialogue management capabilities.

Language Modeling: Utilizing pre-trained language models using neural networks, such as those based on LSTM (Long Short-Term Memory) or transformer architectures. Fine-tune these models on the collected data to enable fluent communication in Sinhala and Tamil languages.

Intent Recognition and Entity Extraction: Implement techniques for intent recognition and entity extraction to understand user queries and extract relevant information, such as account numbers, transaction types, and service requests.

Dialogue Management and System Integration:

Dialogue Flow Design: Design the dialogue flow of the chatbot, mapping out various user interactions and responses. Use techniques like state machines to manage and maintain context throughout the conversation.

Custom Actions: Develop custom Python APIs to retrieve up-to-date information, such as interest rates and exchange rates, by interfacing with backend systems or external APIs.

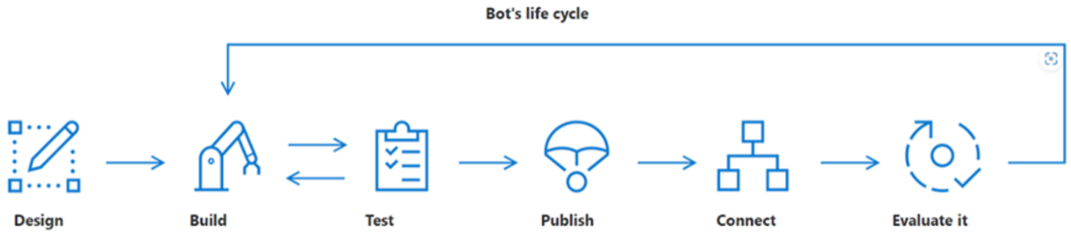
User Authentication and Security: Implement secure user authentication mechanisms using details like mobile numbers, NIC (National Identity Card) numbers, and account numbers, combined with one-time passwords (OTP). Guard against potential security risks like SQL injection by incorporating safeguards.

User Interface and Deployment:

User Interface Development: Create user interfaces for both web and Telegram bot platforms. Existing frameworks like ChainLit and telegram API will be used for this purpose. These interfaces will serve as the front end through which users interact with the chatbot.

Integration with Bank Services: Integrate the chatbot with bank services, allowing users to perform actions such as checking account balances, getting e-statements, and initiating certain transactions.

Data Pipeline and Continuous Integration/Deployment (CI/CD):



Data Pipeline: Implement a data pipeline that encompasses web scraping, data preprocessing, and model training. Automate the flow of data from collection to model updates.

Continuous Integration and Deployment: Utilize GitHub and GitHub Actions to establish a CI/CD pipeline. This will enable seamless integration of new features, updates, and improvements to the chatbot system.

5. Expected Outcomes and Success Criteria:

The ultimate objective is to develop a sophisticated chatbot capable of addressing customer inquiries in Sinhala, Tamil, or English based on individual preferences. These queries pertain to a wide range of topics encompassing banking procedures, available opportunities, and more. For instance, topics span from diverse bank account types and potential loans to associated processes, as well as details regarding fixed deposits.

Moreover, the chatbot will offer supplementary functionalities, aiding customers in navigating personal account operations like checking account balances and seeking transaction histories. Leveraging advanced data science methodologies and techniques, the "Localized Chatbot for Bank Customer Care" project aspires to establish an intelligent, user-centric, and secure chatbot system. This system aims to elevate customer support services for individuals conversing in either Sinhala or Tamil.

The project's success will be gauged by the chatbot's prowess in furnishing precise responses, enhancing customer experiences, and furnishing efficient assistance for a variety of banking-related queries.

6. Preliminary Bibliography:

1. Intelligent Chat Bot for the Banking Sector - A research paper focusing on the development and implementation of an intelligent chatbot in the banking sector.
2. Chatbots in Banking - An online resource discussing the application of chatbots in the banking industry and their potential benefits.
3. How to Build a Chatbot Using Natural Language Processing - A comprehensive guide on building chatbots using natural language processing techniques.
4. How to Create an Intelligent Chatbot in Python Using the spaCy NLP Library - A tutorial demonstrating the creation of an intelligent chatbot using the spaCy NLP library.
5. Rasa Open Source Framework - Documentation for the Rasa open source framework, a tool for building conversational AI.
6. Microsoft Bot Framework - Information about the Microsoft Bot Framework, a platform for building, deploying, and managing chatbots.