Product : AssistBot (A chatbot that provides customer service along with recommending suitable products does upselling)

## **Bot’s Overall goal:**

AssistBot will be a text- based chatbot which will be customer service centric with the goal to provide conversational customer service by engaging the user with interactive questions for general and account related query, as pre sales other banking products based on recommendation system and does upselling.

## **Workflow:**

**Initiating dialogue:**

Our bot will stimulate dialogue with the user and create polite environment to understand the customer query/complain.

**General query:**

It will use a database which has banking(Insurance) FAQ’s and non-banking related question & answers file in json format and user will be able to query with natural language.

**Account related queries:**

Based on the users query the bot will either reply directly with general FAQ’s or ask for the customer Id to further discuss the issue. If user is an existing customer bot will use another database which will have all customer account related info in excel.csv format, from which the bot will answer/ resolve the query using Flask/Django API.

**Bot recommendation:**

Once the query is resolved for the existing customer, based on his/her account details, our bot will recommend Insurance packages or loan products using our recommend model.

**Sales Closing:**

If customer interested in any of the product bot will provide the customer the link to the online application to be filled to get the product and update the customers information to the sales department database for manual sales closing.

**Closing the Chat:**

If the customer is not interested than it will once again ask for any other assistance if required. If yes then the same procedure follows again. If no further assistance required then it will ask the user to rank its assistance on customer satisfaction the scale of 1 to 10. Based on the users input the bot will either offer alternate solution or live agent (if not satisfied) or close the chat.

## **Plan of action(High Level):**

1. **Communication Channel selection** (will be decided from: [*Facebook Messenger, WhatsApp, Web*])
2. **Creating Conversational Architect** using conversational UI’s where customer can send any text and human language, allowing infinite ways to say the same thing with different words. The NLU component will extract the intent and entities, slot and track in to the dialogue management with API to call database request/URL request and then generate message from predefined templates with placeholders.
3. **Creating App’s for integration:** Recommendation model, upselling model which predicts accounts on which upselling can be done based on certain criteria & account model to pull customer details.
4. **Recommendation System:**

Recommendation system will help in catering personalized services to bank customer and prospect.

We will be implementing recommendation system using Santander Bank data set. This data set contains demographic details of the customer and past purchase history (like Credit card, Saving accounts etc.) of the customer.

For existing customer, it will help in cross selling and for new customers it will help in upselling.

Product recommender will suggest Top-N product for the existing customer based on customer ID which will be captured using Chatbot.

For new customer, system will try to understand the taste of the customer by asking few questions related to banking product and based on this input from the customer, it will recommend new product (Basically we will be solving cold start problem here).

1. **Collecting/Preparing Chat Data:**

To start with, we will make a general interaction training dataset in ‘intends’ json format which will be as per the business need for banking and finance domain. The training dataset will be professional English language, as it is for customer service.

Along with the prepared data we will also use chatterbot English and chatbot corpus for training the bot for normal professional language support. (available on Kaggle, data.gov, UCI Machine Learning Repository, etc)

1. **Framework/platform to be used:** Either Microsoft or Chatfuel or without a specific framework using multiple platforms
2. **Implementing the dialogue flow and engineering the NLU:** In this we will bring everything together, the conversational architect, the dialogue flow, the framework selected and the data collected & prepared. The essential task here will be to use these to create a classifier that will map all the incoming text to the systems response.
3. **Testing and deployment:** assessing if the chatbot is capable enough to meet the customers’ needs, do the recommendation, upselling and finally gauge the satisfaction of the customer. Monitoring the entire conversations, collect data, create logs, analyze the data, and keep improving the bot for better conversations.

Author

**Santosh Kamble**

**Tasks:**

1. **Creating Conversational Architect**
2. **Creating App’s for integration**
   1. Account Info Model
   2. Upselling Model
3. **Recommendation Model**
4. **Collecting/Preparing Chat Data**
5. **Implementing the dialogue flow and engineering the NLU – Santosh Kamble TL**

**6 Types of conversational AI:**

1. **Rule based**
2. **Retrieval based**
3. **Generative Method**
4. **Ensemble Methods**
5. **Grounded Learning**
6. **Interactive Learning**

**Chatbot Architecture**

1. **Pattern Matches:** Most of these patterns are structured in Artificial Intelligence Markup Language (AIML). These patterns exist in the chatbot’s database for almost every possible query.
2. **Natural Language Understanding (NLU)**
   1. **Entity**
   2. **Context**
   3. **Intent**
3. **Natural Language Processing (NLP):** [Natural Language Processing](https://www.intellectsoft.net/blog/natural-language-processing-why-the-machines-still-struggle-to-understand-us/) (NLP) makes the chatbot understand input messages and generate an appropriate response. It converts the users’ text or speech data into structured data, which is then processed to fetch a suitable answer. This entire process consists of the following steps
   1. **Tokenization:** The program breaks the string of words into discrete fragments (tokens) that bear a linguistic representation with a unique value for the application.
   2. **Normalization:** It analyzes the text for any possible misspells or typographical errors that could change the intended meaning of users’ queries.
   3. **Entity Recognition:** The chatbot analyzes the text to identify the topic being talked about. It recognizes the entity by searching for a similar category of words, users’ data, or any other required information.
   4. **Dependency Parsing:** The program scrutinizes the text for nouns, verbs, subjects, objects, and phrases to find any dependent information conveyed by the user.
   5. **Sentiment Analysis:** The chatbot software assesses the users’ experiences by analyzing the text. It then forwards the query to a human if needed.
4. **Knowledge Base:** After deciding the intent, the chatbot interacts with the knowledge base to fetch information for the response.

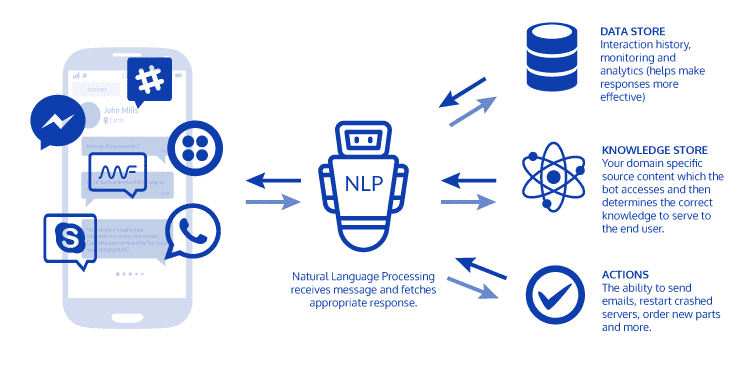
The knowledge base serves as the main response center bearing all the information about the products, services, or the company. It has answers to all the FAQs, guides, and every possible information that a customer may be interested to know.

1. **Data Storage: Another critical component of a chatbot architecture is database storage built on the platform during development. It holds all the chat logs and analytics.**

**Chatbot Architecture Diagram**

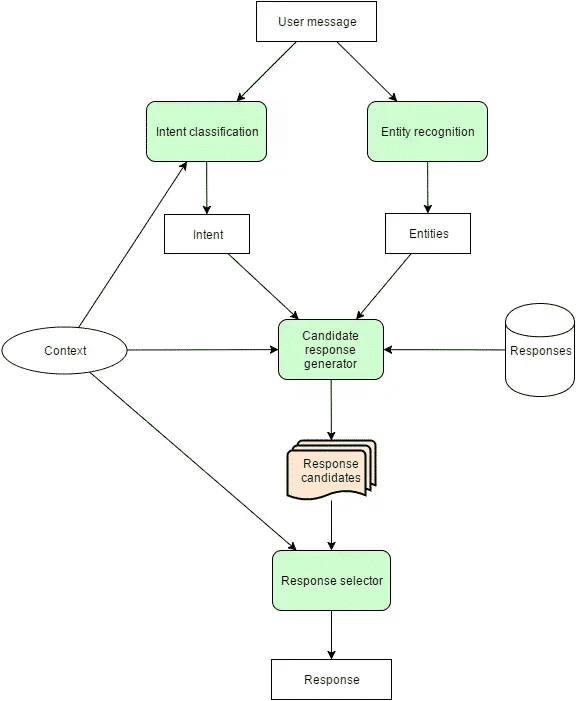
Regardless of how simple or complex a chatbot architecture is, the usual workflow and structure of the program remain almost the same. It only gets more complicated after including additional components for a more natural communication.

Below is the basic chatbot architecture diagram that depicts how the program processes a request.



*Source: Enterprise Bot Manager*

**Whereas, the following flowchart shows how the NLU Engine behind a chatbot analyzes a query and fetches an appropriate response.**



*Source: Enterprise Bot Manager*

**Automatic Speech Recognition (ASR)**

**AKA Speech-To-Text**

Speech Recognition or Speech-To-Text (STT) is a conversion process of turning speech in audio into text.

The goal of ASR is to achieve speaker-independent large vocabulary speech recognition.

it is helpful to give the user a signal to start talking, and keep the utterance as short as possible.

**Background Noise**

The ability to filter out background noise is vital. This background noise interfering with the speaker’s utterance can be traffic, background music, other people speaking etc.

**Microphone**

Microphones can be near-field (like AirPods) or far-field like the Amazon Echo devices. A handset or mobile phone is the microphone in the case of a telephone call. This refers to the ability to process speech at varying distances from microphone.

Factors in speech recognition can be environmental noise, emotional state, fatigue, and distance from microphone.

Hence speech/non-speech segmentation is vital. The ASR system must distinguish between the phonemes (basic unit of speech) that should be recorded for translation vs. the back ground noise.

**Natural Language Understanding**

Natural Language Understanding underpins the capabilities of the chatbot.

Without entity detection and intent recognition all efforts to understand the user come to naught.

Most chatbot architectures consist of four pillars, these are typically *intents*, *entities*, the *dialog flow (State Machine)*, and *scripts*.

The dialog contains the blocks or states a user navigates between. Each dialog is associated with one or more ***intents***and or ***entities***. Session variables can also be employed the decide on which states or nodes must be visited.

The ***intents***and ***entities***constitute the condition on which that dialog is accessed.

The dialog contains the output to the customer in the form of a script, or a message…or wording if you like.