Project Internship Report

on

Chat Bot

By

Sanchit Ghai

Mentor & Guide

A Soundarapandian

For

Hexaware Technologies Limited

SIPCOT IT Park, Siruseri, Chennai

Internship

on

Chat Bot

Period: 15-Jan-2017 to 15-Apr-2018

This report is submitted by **Sanchit Ghai** on 10th APR 2018

Under the Guidance of	
A Soundarapandian, Senior Manager, Business Intelligence and Analytics, Hexaware Technologies Limited, Chennai.	
Pushkhalla Chandramoulli, Software Engineer, Business Intelligence and Analytics, Hexaware Technologies Limited, Chennai.	

Contents

1 EXECUTIVE SUMMARY

ACTIVITY	CHALLENGES IDENTIFIED	CHALLENGES RESOLVED
Client Requirements and Use case:	Creation of the troubleshooting module where bot would check for system errors	The module was dropped due to time constraint
Requirements Gathering & Sample Data Collection, Preprocessing	Removal of ambiguity in the chat script, determining technology used for integration purpose	Ambiguous script was removed/modified, and appropriate technology was identified for integration
Creation of Bot in Dialogflow	Understanding the architecture of Dialogflow	The architecture of Dialogflow was understood.
Integration of Dialogflow with back end	Extraction of parameters from Dialogflow, providing it as input to python, integration with claim center	Parameters extracted through API call as JSON format. Integration with claim center accomplished.
Integration of Python with front end	Determining the technology for integration	Accomplished using REST API, and Flask
Creation of Human Agent Module	Connectivity between bot, user and human agent chat	Accomplished using Memcached
Machine Learning and Classification	Conversion of unstructured data into a form suitable for classification, insufficient data for classification	Challenge were overcome by using keywords as elements for classification, accomplished through repetition of data
Creation of Facebook Messenger Bot	Webhook calls took too long to respond resulting in timeout	Prevention of timeout was accomplished through triggering an intent by making user click on a link
Output accuracy checking	NONE	NONE

2 CHAT BOT

2.1.1 About:

Chat Bot which is also called a chatterbot, bot is a computer program that conducts a conversation through textual method. A chatbot which obtains voice as an input is called voice bot. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Chatbots are typically used in dialog systems for various practical purposes including customer service or information acquisition.

2.1.2 Interaction with a chatbot:

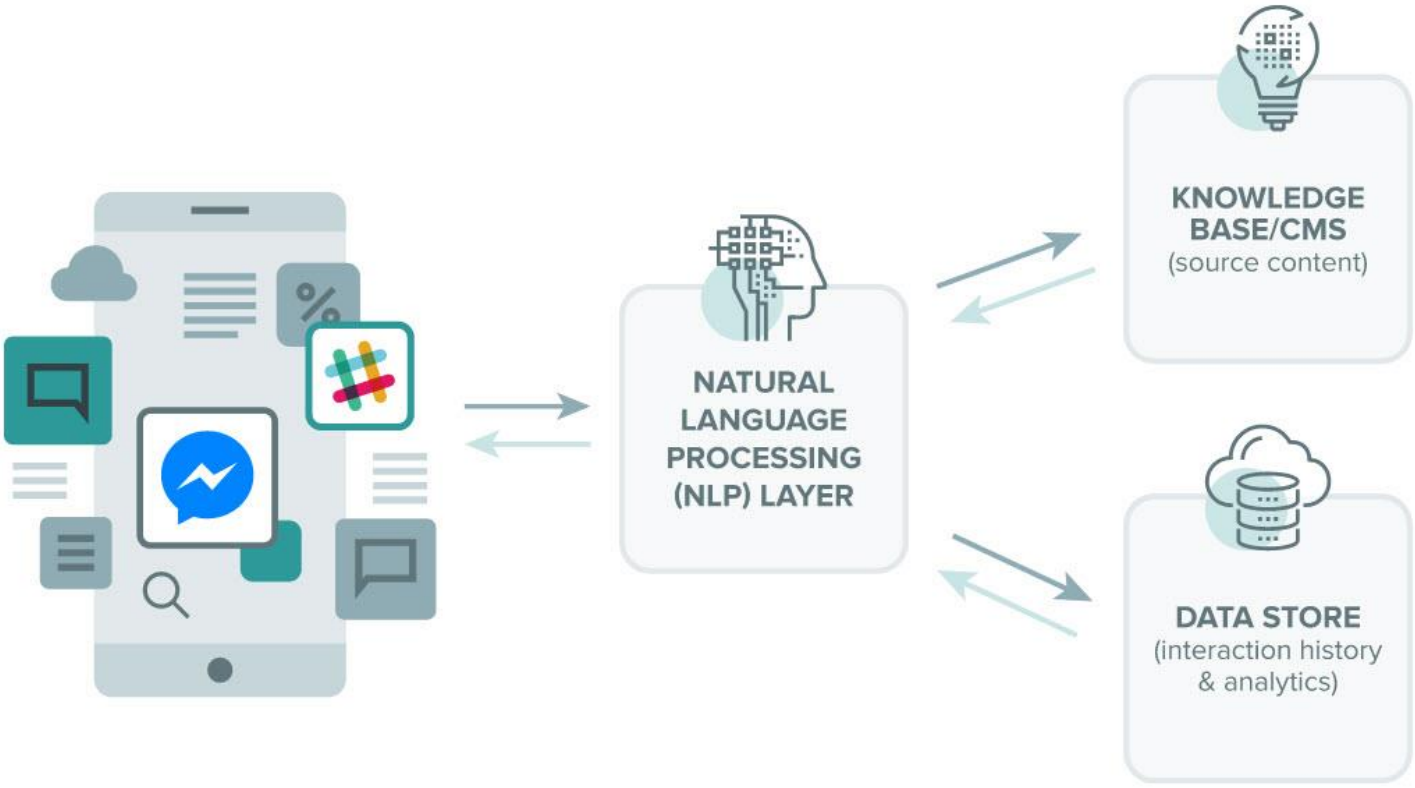
2.1.3 Major Usecases of a chatbot:

- Act as a customer care executive
- NEWS Bot
- Weather Bot
- Automation of a process
- Gather information about user

2.1.4 Objective:

To create a claims intake bot to assist the user with various details about the policy and various FAQ's. The bot will also assist the user to initiate a claim just in case the user wants the bot's help in doing so.

2.1.5 Description



The user provides with the input to the chatbot which can either be the chat window or through Facebook messenger. The input is sent to the NLP module of Dialogflow. If no integration is required, the Dialogflow itself provides the response back. In case if integration is required the input is sent to the python compiler for further processing.

Note: The bot in Dialogflow have to be trained in order to understand the intent in case of different user input for the same intent.

3 REQUIREMENTS

3.1.1 Software prerequisites:

- Python
- Anaconda
- Spyder

3.1.2 Commands to install Python Packages:

- PIP install flask
- PIP install time
- PIP install pandas
- PIP install apiai
- PIP install random
- PIP install requests
- PIP install memcache
- PIP install csv
- PIP install textblob
- PIP install nltk

4 CREATION OF AN INTELLIGENT CHATBOT USING DIALOGFLOW

4.1.1 Dialogflow:

Dialogflow which was formerly (api.ai) is a human computer interaction technology based on Natural Language Processing which is owned and developed by Google. Dialogflow can be used to create a chatbot which after training will answer the question for which it is trained for. Using the Dialogflow API one can easily and seamlessly integrate with various messaging platforms such as Facebook Messenger, Slack etc. Further one of the main advantages of Dialogflow is the machine learning algorithm which trains the bot in such a way that the user can ask a question in several different ways.

4.1.2 Benefits:

- Easily integrate with any third-party messaging apps
- Easy to incorporate with other back end applications through webhook.
- Customizable
- Works well with languages such as python, node js.

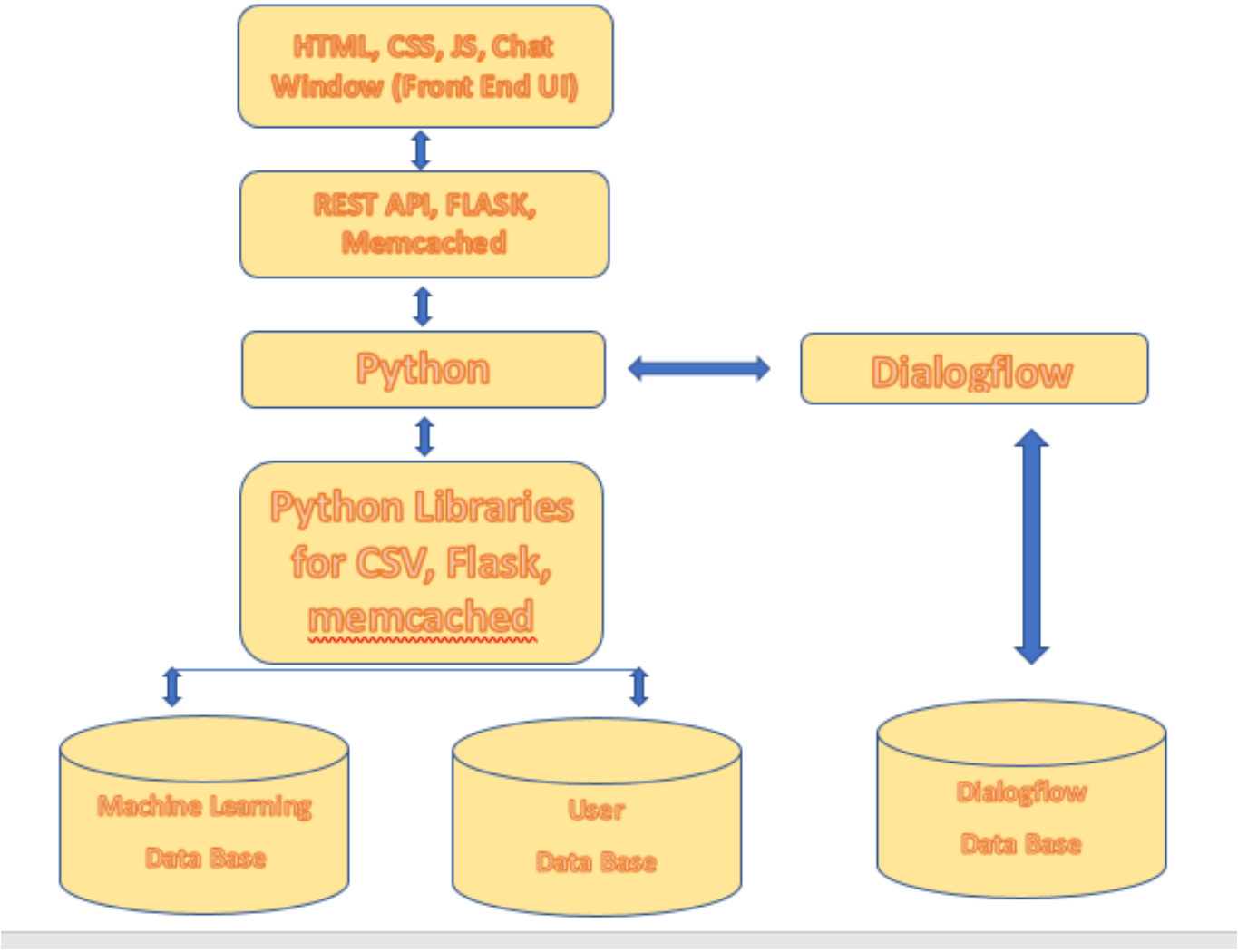
4.1.3 Machine Learning:

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

The Dialogflow uses its own NLP and machine learning algorithm and also gives the developer option to customize the way in which the chat bot is to be trained. Using machine learning the chatbot is able to answer new questions which the bot once was unable to answer.

5 ARCHITECTURE AND FLOW DIAGRAM OF CHATBOT

5.1.1 Architecture:



1. Front End User Interface (Application Layer):

The Application Layer or the User Interface layer consists of HTML, CSS, JS which are used in the creation of the chat windows of Human Agent and the User. Whenever the user enters some text into the chat window it is then sent to python code to identify the intent of the question to provide proper response back to the user in the chat window. The human agent chat window displays the entire conversation between the user and the bot which is also used to provide the answer if the intent of the question is not identified.

1.1 HTML:

HTML or (Hypertext Markup Language) is used for the creation of a webpage which is used for the creation of the front end chat windows through which the chat application accepts and displays the texts

1.2 CSS:

CSS or (Cascading Style Sheet) is a document which accompanies HTML which is responsible for adding designs and formatting of the HTML page so that the contents of the page are more presentable and customizable.

1.3 JS:

JS or (Java Script) as the name suggests is a script which is added along with the HTML page for the addition of new functionalities and to make the page interactive such as drop downs, menus etc.

2. API and Framework Layer:

API or (application programming interface) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. The API layer consists of various APIs which helps python to push contents into the front end webpage. The API and framework layer consists of the REST API, FLASK, Memcached.

2.1 FLASK:

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. Flask is responsible for the creation of the REST API.

2.2 REST API:

Rest API or (REpresentational State Transfer) or RESTful web services provide interoperability between computer systems on the Internet. REST-compliant web services allow the requesting systems to access and manipulate textual representations of web resources by using a uniform and predefined set of stateless operations. Other kinds of web services, such as WSDL and SOAP, expose their own arbitrary

sets of operations. The Rest API interacts with the Java Script and initiates an array of array of request-response operations which assists in pushing the text from python to the chat window.

2.3 Memcached:

Memcached is a general-purpose distributed memory caching system. Memcached is used to store the text from the user, human agent, bot which is either used to display the text the human agent writes in the chat window into the user chat window and vise versa or to save the entire conversation into the database from which the machine learning algorithm is run so that any unknown question asked for the second time can be answered for the second time.

3. Python Layer:

Python is a interpreted high level programing language for general purpose programming. Python obtains the conversation from the user communicates with the dialog flow either to get the response or to identify the intent so that it can obtain the response from the integration section. Python is also responsible for maintaining the CSV files which consists of the user, policy, claim creation data and training dataset and test dataset used for running the Machine Learning Algorithm. The machine learning algorithm is also written in python which runs the algorithm.

4. Libraries:

A library is a collection of non-volatile resources used by computer programs, often to develop software. These may include configuration data, documentation, help data, message templates, pre-written code and subroutines, classes, values or type specifications.
Various libraries such as the CSV library, flask library, Memcached libraries have been imported into the python code so that various resources related to them can be automatically imported into the code every-time the code is run.

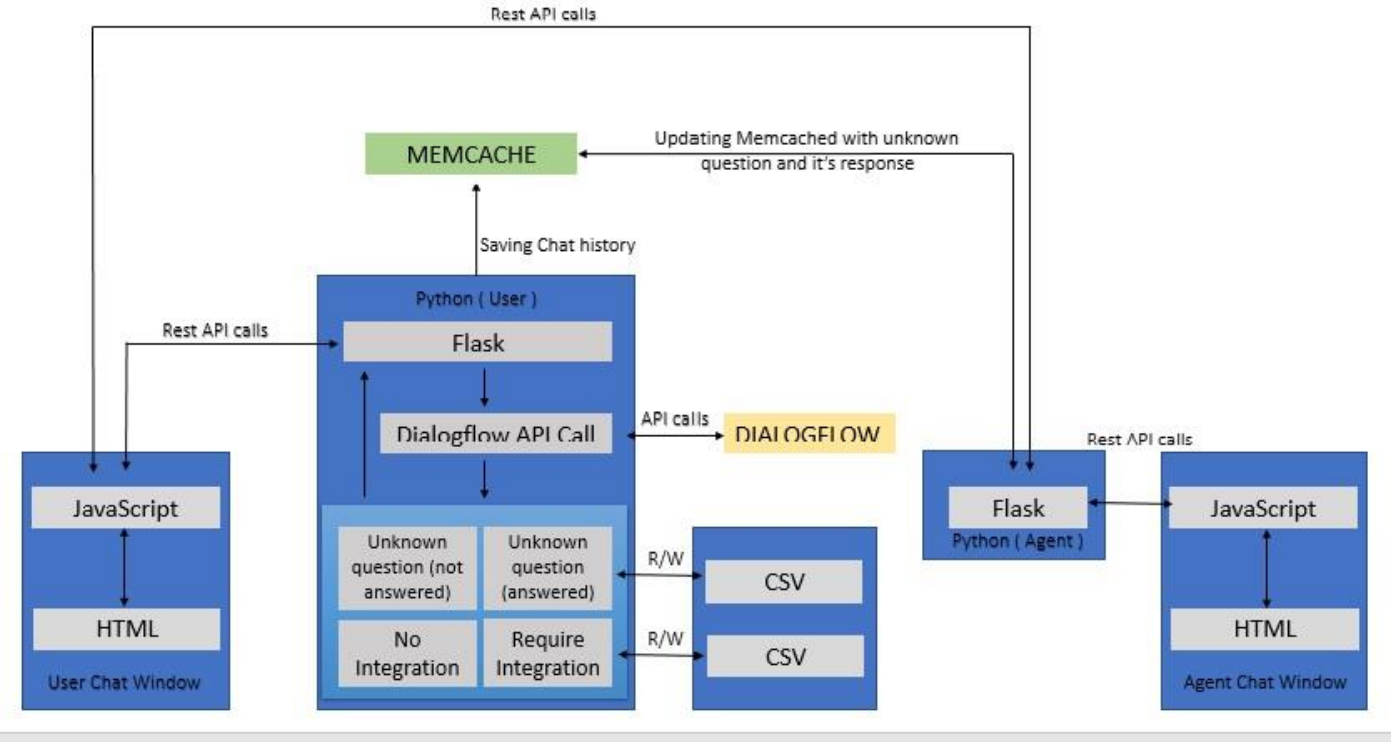
5 Storage Layer:

The storage layer consists of databases in which various details which are required by the application. Here the use of CSV files has been made to replicate the functionalities of the database. Python is responsible to read and write information from and into the csv file. The storage layer stores the user data, training and test dataset which is used by python.

6 Dialogflow:

Dialogflow is a Google-owned developer of human–computer interaction technologies based on natural language conversations. Dialogflow is used to provide the user input to determine the intent of the user input. The intent is recognized by the text classification algorithm google uses to train the bot. The bot is created in dialog flow specifying the intents along with the training sentences and entities.

5.1.2 Flow Diagram:



- **Scenario 1: No Integration.**

  Input: User Query

  Output: Response from Dialogflow.

  Sample Flow:
  Chat Window → JavaScript (User) → Flask (User) → Dialogflow API → Speech → Flask → JavaScript → Chat Window

  Description: Once the user has provided the input in the chat window, the query is sent to the JavaScript which then sends the query to flask via the REST API then the query hits the Dialogflow API to match for intent and obtain the respective response or speech. The Dialogflow API sends the speech to flask which then sends it to the JavaScript which appends the speech to the chat window.

- **Scenario 2: Integration Required.**

  Input: User Query

  Output: Response from Python.

  Sample Flow:
  Chat Window → JavaScript (User) → Flask (User) → Dialogflow API → Parameters → Read/Write CSV → Speech → Flask → JavaScript → Chat Window

  Description: Once the user has provided the input in the chat window, the query is sent to the JavaScript which then sends the query to flask via the REST API then the query hits the Dialogflow API to match for intent once the intent is obtained the Dialogflow sends the parameters to python using webhook which then either performs a task such as create a claim and return the claim number or looks up the user data from the csv file by performing read/write operations and returns the appropriate output. The returned output is saved in speech and the python sends the speech to flask which then sends it to the JavaScript which appends the speech to the chat window
  .

- **Scenario 3: Answering Unknown Question.**

  Input: User Query Unknown to Bot

  Output: Response from Human Agent.

  Sample Flow:
  1 Chat Window → JavaScript (User) → Flask (User) → Dialogflow API → Unknown Question not answered → memcached → Flask(Agent) → JavaScript(Agent) → Agent Input → Memcached → Flask(Agent) → JavaScript(User) → Chat Window
  2 Memcached → CSV(Write)

  Description: Once the user has provided the input in the chat window, the query is sent to the JavaScript which then sends the query to flask via the REST API then the query hits the Dialogflow API to match for intent once the intent cant be determined it goes to the fallback stage the Dialogflow uses webhook sends the parameters to python. The python initiates the memcached hits the flask of the human agent which then hits the REST API of human agent which then goes to the JavaScript of the human agent. When the human agent provides with the response which is then sent to the flask of human agent which then hits memcached which then sends the speech or response to the JavaScript of the user which then appends the speech to the user chat window.

- **Scenario 4: Answering Unknown Question already answered.**

  Input: User Query Unknown to Bot

  Output: Response from Python.

  Sample Flow:
  1 Chat Window → JavaScript (User) → Flask (User) → Dialogflow API → Unknown Question already answered → Read CSV → Flask → JavaScript(User) → Chat Window

  Description: Once the user has provided the input in the chat window, the query is sent to the JavaScript which then sends the query to flask via the REST API then the query hits the Dialogflow API to match for intent once the intent can't be determined it goes to the fallback stage the Dialogflow uses webhook sends the parameters to python. Python reads the CSV file to check if the question has already been answered. Then python finds the answer to the requested question in the CSV file. The answer is returned and saved inside speech. The speech is then sent to flask which then sends it to the JavaScript which appends the speech to the user chat window.

6 CONCLUSION

OBJECTIVE	To make a Chatbot who can respond to all the Claim Queries of user and learn to answer an unknown question once answered by Human Agent.
SERVICE	DIALOGFLOW
MACHINE LEARNING	Deep Learning
ACHIEVEMENT	The Chatbot was able to satisfy all the criteria and parameters successfully.