Applied Machine Learning

Project Report

**COVID-19 Chatbot**

Ronak Kumar

Nitin Kumar

**Problem Statement**

The fundamental reason for the plan is to build the language gap between the client and health providers by giving prompt answers to the questions asked by the users. Presently individuals are almost certainly dependent on the web. They stay away from going to the emergency clinic for little issues which might turn into a significant infection in future. Using question answer intents is becoming an easy way to answer questions instead of browsing relevant documents on the web.

Chatbot is an Entity which imitates human conversation in its specific acknowledged set-up along with a text or vocal language with procedures, for example, Natural Language Processing (NLP). The point of this framework is to repeat an individual's conversation. The improvement of chatbot applications should be possible with making a UI to send input and get reactions. Medical chatbots can be created by utilising artificial algorithms that examine a user's inquiries and remember them and give answers to related inquiries. Our aim is to build a chatbot expert system which can identify the intent of the input message and respond to COVID-19 user-queries and frequently asked questions. Hopefully, the deployed application can be helpful for people fighting in the COVID pandemic.

**Analysis of Technologies**

In the current market, there are a number of technological frameworks that can be used to build chatbots. Some of the following are listed below :-

* Rasa: It is an open-source framework involving two significant components which are Rasa NLU and Rasa Core. Rasa stack NLU is responsible for natural language understanding whereas Rasa core helps developers create intelligent, conversational chatbots. It requires minimal training data and so chatbots can be prototyped very easily and quickly. Here are some of the advantages and disadvantages of using Rasa :-

| Advantages | Disadvantages |
| --- | --- |
| Deployment on your personal server by keeping all the components in-house. | Not suitable for beginners |
| Highly customizable and flexible to allow developers create chatbot with desired features |  |
| Allows multiple environments for development, staging, and production |  |
| Offers analytics support for various data that allows us to understand customers better |  |
| Works on interactive learning hence it keeps learning on its own as it interacts with people |  |

* Wit AI: It is a free chatbot framework and allows developers create all the intents and the entities. It is an open-source project where developers can easily create bots with human-level intelligence without teaching the bot the basics.

| Advantages | Disadvantages |
| --- | --- |
| Easy deployment | Hard to retrieve missing parameters |
| Open-source allowing more community support |  |
| The best bot-building tool |  |
| Multiple SDK support with different programming language |  |
| Supports around 80+ languages from around the world |  |

* DialogFlow: An AI chatbot framework that provides machine learning capabilities, and inbuilt NLP features and integrations with other popular platforms used for communication. Using this, developers create highly intelligent chatbots that use Google’s machine learning models and understand the language.

| Advantages | Disadvantages |
| --- | --- |
| Supports both voice-based and text-based assistants | Fine control over dialogue processing not available |
| Easy to understand and learn, even for beginners |  |
| Best quality and rich conversations |  |
| Multiple SDK support with different programming language |  |
| Supports around 20+ languages from around the world |  |

* IBM Watson: One of the most widely known AI chatbot platforms used by developers. It offers different types of bot-building tools and has built-in machine learning capability. IBM Watson stands out in terms of integration, features and customization.

| Advantages | Disadvantages |
| --- | --- |
| Highly advanced and capable ML engine | Too many tools can be confusing |
| Allows automated predictive analysis |  |
| Doesn’t collect data that you use for building the bot and stores on private cloud, for maximum security and confidentiality |  |
| Visual recognition security |  |
| Supports around 10+ languages from around the world |  |

* Amazon Lex: A versatile chatbot framework using complicated bot-building tools, Its a great AI chatbot framework for beginners.

| Advantages | Disadvantages |
| --- | --- |
| Automated speech recognition and conversion to text | Language barrier to develop an artificial intelligence chatbot |
| Integrated with AWS, it is highly scalable |  |
| Multiple SDK support |  |
| Provides SDK for many different platforms |  |

**Methodology**

We chose to build our own chatbot due to the flexibility of intents and responses that we can achieve using that. Our project code includes an intents.json file in which we can customise our own responses instead of depending on the framework to provide. Also, we tried to implement Deep Learning using LSTM so as to learn the intents file and predict the responses.

We have used two different python files app.py, chatbot\_backend.py with an intents.json file that reads the intents file and runs the Bi-LSTM model behind. We have used the Streamlit UI and Chat library for building the front end of the webapp.