**CHAT BOT NOTES**

What is RASA ?

It’s a python framework used for building chat bots.

What is a framework ?

A framework is a collection of libraries and APIs to help you with building an application. JAVA Collections is a framework containing various APIs/interfaces to various types of objects to act as collections.

What is a library ?

A Library is a collection of functions /objects that serves one particular purpose. It’s one of the purpose is code reusability.

What is API and how is it different from library ?

An API is an interface for other programs to interact with your program without having direct access. Unlike libraries, API allows interaction between two layers of a program securely where as Libraries are just for code reusability purpose.

In other words, APIs and libraries are a part of framework. Framework acts as skeleton to our code and our application works within a framework, i.e. control is with framework and not with application where as in case of simply using a library or API interface in our application, control of execution is with our application.

RASA STACK

Rasa stack is a collection of two open source machine learning libraries RASA NLU and RASA CORE.

RASA NLU (Natural Language Understanding) analyzes the question to specify intent and extract the entity. Then a dialogue management model trained with RASA CORE will predict appropriate response. To send a response it may also use API .

What is Tensor flow ?

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks.

It contains machine learning and deep learning models and algorithms.

Procedure:

Started with: Installing anaconda with python 3.7.6. . Although latest anaconda is with python 3.8.3 but tensor flow doesn’t support it yet.

Then opened the anaconda prompt and navigated to the folder where I will create the project.

* In anaconda prompt , in order to switch to D:/ from C:/ I simply have to write d:/ and press enter. No need to write cd d:/

Then we create a python environment for our chat bot with command **conda create –name chatBotEnv python 3.7.6** and then activate that env.

ujson is another dependency we install as part of this environment suing command **conda install ujson.**

**Before installing rasa you want to make sure that you install visual c++ redistributable 14 or higher.**

Then we install rasa using command **pip install rasa**

Then we can switch to the chatBot environment by using command **conda activate chatBotEnv**

If you don’t remember the env name you can use **conda list env** command.

Command **code .** will open up the current directory with vs code and all the editable files will be available.

Comand **rasa shell** starts the bot.

When we change the nlu or domain files then we have to retrain the bot and a new rasa model is made which will be used for the bot. By default attest model is used when you start the bot by rasa shell command.

Then in the end, inside my .html file above closing body tag i.e. <\body> I paste the following code.

Below, we need to change 3 things. Socket url, title and subtitle.

SocketUrl is the url of the website on which rasa chatbot is live.

</div>

<div id="webchat"/>

<script src="https://storage.googleapis.com/mrbot-cdn/webchat-0.4.1.js"></script>

<script>

    WebChat.default.init({

        selector: "#webchat",

        initPayload: "/get\_started",

        interval: 300, // 1000 ms between each message

        customData: {"userId": "123"}, // arbitrary custom data. Stay minimal as this will be added to the socket

        socketUrl: "http://localhost:5005",

        socketPath: "/socket.io/",

        title: "Jarvis",

        subtitle: "My First Bot ever",

        profileAvatar: "http://to.avat.ar",

    })

</script>

Then I edit the credentials.yml file in my chatBotEnv locaction and uncomment the socketIO code as follows,

socketio:

user\_message\_evt:user\_uttered

bot\_message\_evt:bot\_uttered

session\_persistence:true

Then to run the chatbot in the website we run the chatbot server by using command,

**Rasa run –cors “\*” –debug**

To stop the server we press, **ctrl+PB** . It means ctrl+ pause break.

Some Resources:

<https://www.analyticsvidhya.com/blog/2019/04/learn-build-chatbot-rasa-nlp-ipl/>

https://github.com/botfront/rasa-webchat