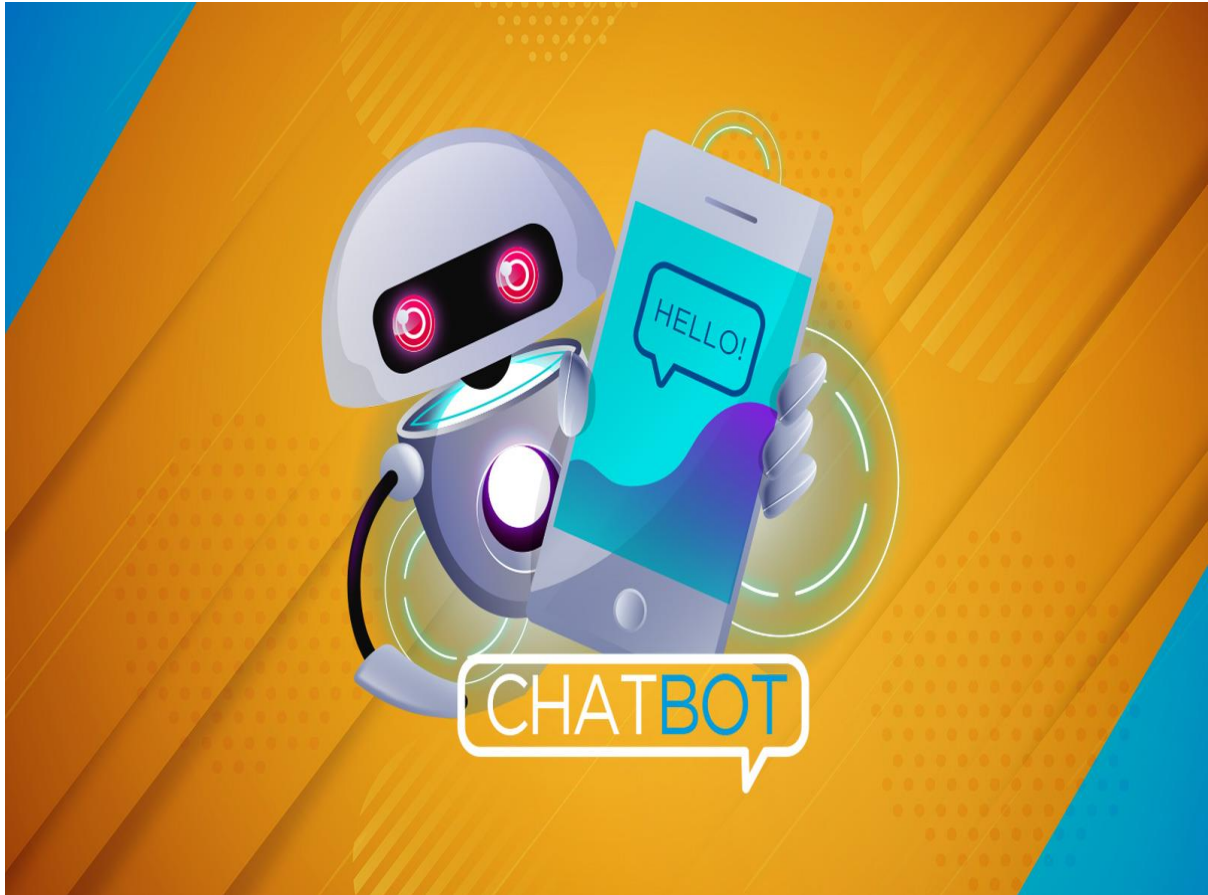


CAREERBOT USING RASA FRAMEWORK



BY- APARNA

EMAIL- aparnaagust3@gmail.com

TABLE OF CONTENTS

SCOPE OF WORK-----	3
INTRODUCTION-----	4
WORKING OF CHATBOT-----	5
CHATBOT FRAMEWORKS-----	6
RASA OVERVIEW-----	7
RASA ARCHITECTURE-----	8
RASA, RASA X AND RASA ENTERPRISE-----	9
PREREQUISITES-----	11
INSTALLATION OF RASA X-----	11
FILES MADE BY RASA INIT-----	17
CHOOSING A PIPELINE-----	20
IMPLEMENTATION-----	21
TELEGRAM INTEGRATION-----	28
RESULT AND OUTCOMES-----	31
CONCLUSION-----	35
REFERENCES-----	36

SCOPE OF WORK

The Scope of the project consists of developing a Chatbot which is trained to answer career related queries and questions.

Scope also includes:

- Developing set of scenarios, questions and sentences for training the chatbot so as to develop chatbot's vocabulary
- Configuring the Chatbot using the Rasa framework
- Integration and testing of the Chatbot using Telegram

Tasks that this chatbot is able to do includes:

- Guidance for best career options that one should pursue
- Finds jobs according to user's interests
- Answers queries related to exams that one should take

INTRODUCTION

A **chatbot** is a computer program that simulates human conversation through voice commands or text chats or both. Chatbot, short for chatterbot, is an Artificial Intelligence (AI) feature that can be embedded and used through any major messaging applications. There are a number of synonyms for chatbot, including "talkbot," "bot," "IM bot," "interactive agent" or "artificial conversation entity."

A chatbot can be either a simple rule-based engine or an intelligent application leveraging Natural Language Understanding. Many organizations today have started using chatbots extensively. Chatbots are becoming famous as they are available 24*7, provide a consistent customer experience, can handle several customers at a time, are cost-effective and hence, results in a better overall customer experience.

Components of Chatbot

1. Intents

- Intents capture user goal.
- Examples:

Hello

Hi

These are both examples of greetings so intent is greetings

2. Entities

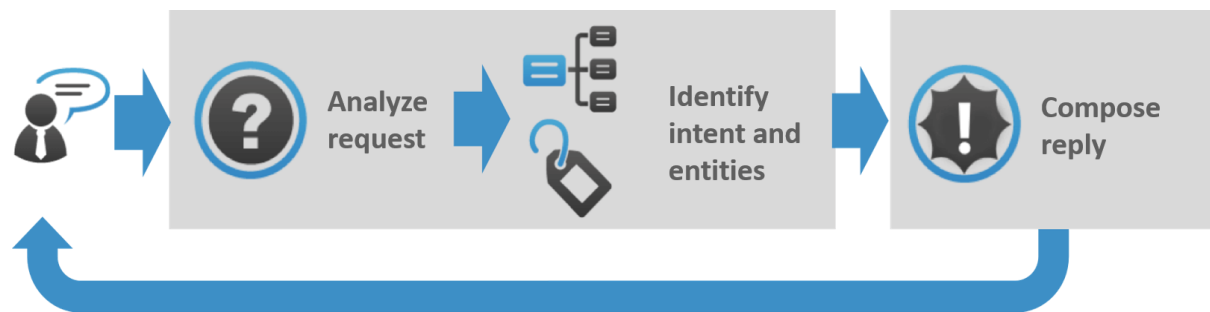
- Entities capture specific values in user input.
- Example: When is your Delhi store open?
- Delhi is a specific value that is a location so entity is Location

3. Dialog or Response

WORKING OF CHATBOT

There are two different tasks at the core of a chatbot:

- 1) User request analysis
- 2) Returning the response



1) User request analysis: this is the first task that a chatbot performs. It analyzes the user's request to **identify the user intent** and to **extract relevant entities**.

2) Returning the response: once the user's intent has been identified, the chatbot must provide the most appropriate response for the user's request.

USES OF CHATBOTS

- Customer support
- Frequently Asked Questions
- Addressing Grievances
- Appointment Booking
- Automation of routine tasks
- Address a query

CHATBOT FRAMEWORKS

Below mentioned are few frameworks which are being used to make chatbot:

1. IBM Watson Conversation
2. Microsoft Bot Framework
3. Dialogflow
4. Amazon Lex
5. ManyChat
6. RASA

In this project I have used **RASA** as the framework



RASA OVERVIEW

Rasa is an open source machine learning framework for building contextual AI assistants and chatbots.

Rasa has two main modules:

1. **Rasa NLU** for understanding user messages
2. **Rasa Core** for holding conversations and deciding what to do next

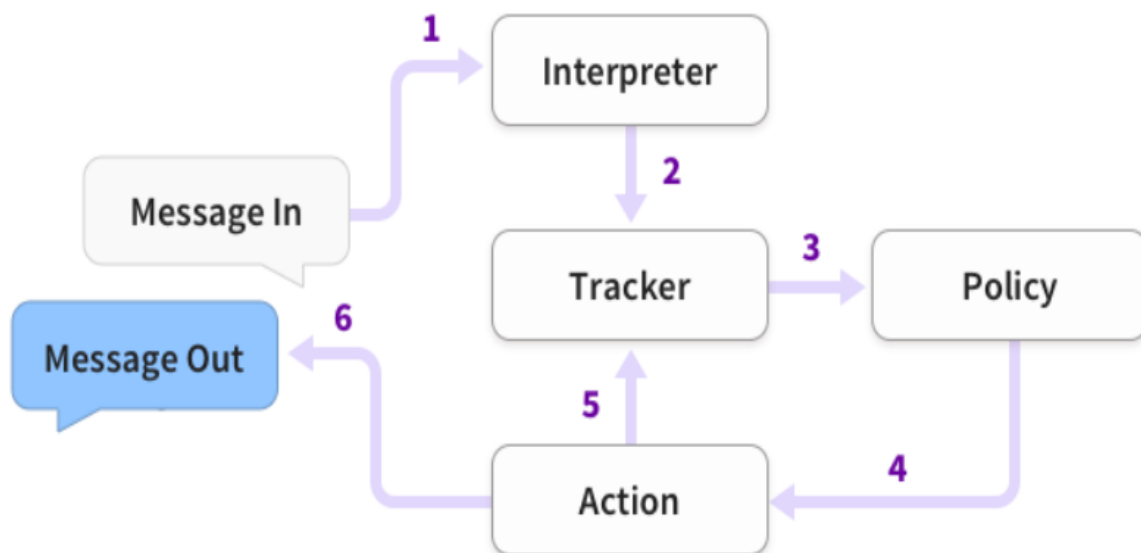
Rasa NLU

This is the place, where rasa tries to understand User messages to detect **Intent** and **Entity** in your message. Rasa NLU has different components for recognizing intents and entities.

Rasa Core

This is the place, where Rasa try to help you with contextual message flow. Based on User message, it can predict dialogue as a reply and can trigger Rasa Action Server.

RASA ARCHITECTURE



1. The message is received and passed to an Interpreter, which converts it into a dictionary including the original text, the intent, and any entities that were found. This part is handled by NLU.
2. The Tracker is the object which keeps track of conversation state. It receives the info that a new message has come in.
3. The policy receives the current state of the tracker.
4. The policy chooses which action to take next.
5. The chosen action is logged by the tracker.
6. A response is sent to the user.

RASA , RASA X AND RASA ENTERPRISE

Rasa Open Source

Rasa Open Source is a conversational AI framework for building contextual assistants.

Rasa Open Source includes

- **NLU**: determines what the user wants and captures key contextual information
- **Core**: selects the next best response or action based on conversation history
- **Channels and integrations**: connect assistant to users and backend systems

Rasa X

It is a toolset to make it review and improve AI assistants

Running on top of Rasa open source, Rasa X strengthens and extend its features allowing developers to :

- Enable and review conversations
- Improve assistant based on learnings
- Deploy new updates to users

Deploy your minimum viable assistant on a server and improve it using Rasa X



Rasa Enterprise

It is an integrated platform that includes Rasa Open Source + RasaX

Plus:

- Analytics
- Multiple deployment environments and many more

Unlike Rasa and Rasa X it is not free

I have used Rasa X in this project

PREREQUISITES

The prerequisites for developing and understanding a chatbot using Microsoft Azure are:

- Python installed
- Microsoft Build tools with visual c++ 14.0 installed.

Link: <https://visualstudio.microsoft.com/downloads/>

I have made this project using Anaconda and Visual C++

INSTALLATION OF RASA X

1. Make a folder where chatbot will be saved.
2. Open Anaconda Prompt
3. Enter **cd <path where chatbot will be saved>**

For example: `cd C:\Users\aparn\OneDrive\Documents\Project\Careerbot`

4. Enter **conda create --name <name of environment that you want>python==<version>**

For example: `conda create --name rasa python==3.7.6`

```
Anaconda Prompt (anaconda3) - conda install ujson - conda install tensorflow
The following NEW packages will be INSTALLED:

ca-certificates    pkgs/main/win-64::ca-certificates-2020.1.1-0
certifi            pkgs/main/win-64::certifi-2020.6.20-py37_0
openssl            pkgs/main/win-64::openssl-1.1.1g-he774522_0
pip                pkgs/main/win-64::pip-20.1.1-py37_1
python             pkgs/main/win-64::python-3.7.6-h60c2a47_2
setuptools         pkgs/main/win-64::setuptools-47.3.1-py37_0
sqlite             pkgs/main/win-64::sqlite-3.32.3-h2a8f88b_0
vc                 pkgs/main/win-64::vc-14.1-h0510ff6_4
vs2015_runtime     pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_2
wheel              pkgs/main/win-64::wheel-0.34.2-py37_0
wincertstore       pkgs/main/win-64::wincertstore-0.2-py37_0
zlib               pkgs/main/win-64::zlib-1.2.11-h62dcd97_4

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
```

After entering this command, it will install some packages and everywhere give a yes or y.

5. Now enter: **conda activate<name of environment that was kept in step 4 >**

This command will activate the environment of project

6. Now: **conda install ujson**

ujson module allows to convert between Python objects and the JSON data format.

7. Then, **conda install tensorflow**

It will install Tensorflow

After executing these commands, we get a snapshots just like these:

```
(base) C:\Users\aparn\OneDrive\Documents\Project\Careerbot>conda activate installingrasa

(installingrasa) C:\Users\aparn\OneDrive\Documents\Project\Careerbot>conda install ujson
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.8.2
  latest version: 4.8.3

Please update conda by running

  $ conda update -n base -c defaults conda
```

```
(installingrasa) C:\Users\aparn\OneDrive\Documents\Project\Careerbot>conda install tensorflow
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.8.2
  latest version: 4.8.3

Please update conda by running

  $ conda update -n base -c defaults conda


## Package Plan ##

environment location: C:\Users\aparn\anaconda3\envs\installingrasa

added / updated specs:
- tensorflow

The following packages will be downloaded:
```

```
Anaconda Prompt (anaconda3) - conda install ujson - conda install tensorflow
- tensorflow
```

The following packages will be downloaded:

package	build	
-----	-----	
scipy-1.5.0	py37h9439919_0	11.8 MB
-----	-----	
Total:		11.8 MB

The following NEW packages will be INSTALLED:

_tfflow_select	pkgs/main/win-64::_tfflow_select-2.2.0-eigen
absl-py	pkgs/main/win-64::absl-py-0.9.0-py37_0
astor	pkgs/main/win-64::astor-0.8.0-py37_0
blas	pkgs/main/win-64::blas-1.0-mkl
blinker	pkgs/main/win-64::blinker-1.4-py37_0
brotlipy	pkgs/main/win-64::brotlipy-0.7.0-py37he774522_1000
cachetools	pkgs/main/noarch::cachetools-4.1.0-py_1
cffi	pkgs/main/win-64::cffi-1.14.0-py37h7a1dbc1_0
chardet	pkgs/main/win-64::chardet-3.0.4-py37_1003
click	pkgs/main/noarch::click-7.1.2-py_0
cryptography	pkgs/main/win-64::cryptography-2.9.2-py37h7a1dbc1_0
gast	pkgs/main/win-64::gast-0.2.2-py37_0
google-auth	pkgs/main/noarch::google-auth-1.14.1-py_0
google-auth-oauth~	pkgs/main/noarch::google-auth-oauthlib-0.4.1-py_2
google-pasta	pkgs/main/noarch::google-pasta-0.2.0-py_0
grpcio	pkgs/main/win-64::grpcio-1.27.2-py37h351948d_0
h5py	pkgs/main/win-64::h5py-2.10.0-py37h5e291fa_0
hdf5	pkgs/main/win-64::hdf5-1.10.4-h7ebc959_0
icc_rt	pkgs/main/win-64::icc_rt-2019.0.0-h0cc432a_1
idna	pkgs/main/noarch::idna-2.9-py_1
intel-openmp	pkgs/main/win-64::intel-openmp-2020.1-216
keras-applications	pkgs/main/noarch::keras-applications-1.0.8-py_0
keras-preprocessi~	pkgs/main/noarch::keras-preprocessing-1.1.0-py_1
libprotobuf	pkgs/main/win-64::libprotobuf-3.12.3-h7bd577a_0
markdown	pkgs/main/win-64::markdown-3.1.1-py37_0
mkl	pkgs/main/win-64::mkl-2020.1-216
mkl-service	pkgs/main/win-64::mkl-service-2.3.0-py37hb782905_0
mkl_fft	pkgs/main/win-64::mkl_fft-1.1.0-py37h45dec08_0
mkl_random	pkgs/main/win-64::mkl_random-1.1.1-py37h47e9c7a_0
numpy	pkgs/main/win-64::numpy-1.18.5-py37h6530119_0
numpy-base	pkgs/main/win-64::numpy-base-1.18.5-py37hc3f5095_0
oauthlib	pkgs/main/noarch::oauthlib-3.1.0-py_0
opt_einsum	pkgs/main/noarch::opt_einsum-3.1.0-py_0
protobuf	pkgs/main/win-64::protobuf-3.12.3-py37h33f27b4_0
pyasn1	pkgs/main/noarch::pyasn1-0.4.8-py_0
pyasn1-modules	pkgs/main/noarch::pyasn1-modules-0.2.7-py_0
pyparser	pkgs/main/noarch::pyparser-2.20-py_0

8. Now install rasa x using following command:

pip install rasa-x --extra-index-url <https://pypi.rasa.com/simple>

Anaconda Prompt (anaconda3) - conda install ujson - conda install tensorflow

```
(installingrasa) C:\Users\aparn\OneDrive\Documents\Project\Careerbot>pip install rasa-x --extra-index-url https://pypi.rasa.com/simple
Looking in indexes: https://pypi.org/simple, https://pypi.rasa.com/simple
Collecting rasa-x
  Downloading https://pypi.rasa.com/api/package/rasa-x/rasa_x-0.29.3-py3-none-any.whl (1.6 MB)
    |#####| 1.6 MB 6.4 MB/s
Collecting alembic<2.0.0,>=1.0.10
  Downloading alembic-1.4.2.tar.gz (1.1 MB)
    |#####| 1.1 MB 1.3 MB/s
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing wheel metadata ... done
Collecting python-dateutil<2.9,>=2.8
  Using cached python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Requirement already satisfied: setuptools<=41.0.0 in c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages (from rasa-x) (47.3.1.post20200622)
Collecting ruamel.yaml<0.17,>=0.16
  Using cached ruamel.yaml-0.16.10-py2.py3-none-any.whl (111 kB)
Collecting jsonschema<4.0,>=3.2
  Using cached jsonschema-3.2.0-py2.py3-none-any.whl (56 kB)
Collecting rasa<2.0.0,>=1.9.5
  Using cached rasa-1.10.3-py3-none-any.whl (510 kB)
Requirement already satisfied: requests<3.0,>=2.23 in c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages (from rasa-x) (2.24.0)
Requirement already satisfied: cryptography<3.0,>=2.7 in c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages (from rasa-x) (2.9.2)
Collecting ujson<2.0,>=1.35
  Downloading ujson-1.35.tar.gz (192 kB)
    |#####| 192 kB ...
Collecting sanic<20.0.0,>=19.12.2
  Using cached sanic-19.12.2-py3-none-any.whl (72 kB)
Collecting sanic-jwt<1.4.0,>=1.3.2
  Downloading sanic-jwt-1.3.2.tar.gz (18 kB)
Collecting psycogp2-binary<3.0.0,>=2.8.2
  Using cached psycogp2_binary-2.8.5-cp37-cp37m-win_amd64.whl (1.1 MB)
Collecting questionnaire<1.6.0,>=1.5.1
  Using cached questionnaire-1.5.2-py3-none-any.whl (26 kB)
Collecting isodate<0.7,>=0.6
  Downloading isodate-0.6.0-py2.py3-none-any.whl (45 kB)
    |#####| 45 kB 3.4 MB/s
Collecting GitPython<4.0.0,>=3.0.8
  Downloading GitPython-3.1.3-py3-none-any.whl (451 kB)
    |#####| 451 kB 6.4 MB/s
Collecting attrs<20.0,>=19.3
  Using cached attrs-19.3.0-py2.py3-none-any.whl (39 kB)
Collecting kafka-python<2.0,>=1.4
  Using cached kafka_python-1.4.7-py2.py3-none-any.whl (266 kB)
Collecting aiohttp<4.0,>=3.6
  Using cached aiohttp-3.6.2-cp37-cp37m-win_amd64.whl (649 kB)
Collecting pika<2.0.0,>=1.1.0
  Using cached pika-1.1.0-py2.py3-none-any.whl (148 kB)
Collecting sanic-cors<0.11.0,>=0.10.0.post3
  Using cached Sanic_Cors-0.10.0.post3-py2.py3-none-any.whl (17 kB)
```

9. Now install spacy. **spaCy** is a free open-source library for Natural Language Processing in **Python**.

pip install spacy

10. Enter the command for rasa init

The **rasa init** command creates all the files that a **Rasa** project needs and trains a simple bot on some sample data.

rasa init

```
(installingrasa) C:\Users\aparn\OneDrive\Documents\Project\Careerbot>rasa init
Welcome to Rasa! 🤖 🤖

To get started quickly, an initial project will be created.
If you need some help, check out the documentation at https://rasa.com/docs/rasa.
Now let's start! 🤖 🤖 🤖🤖

? Please enter a path where the project will be created [default: current directory] .
Created project directory at 'C:\Users\aparn\OneDrive\Documents\Project\Careerbot'.
Finished creating project structure.
? Do you want to train an initial model? 🤖 🤖 🤖🤖🤖 Yes
Training an initial model...
Training Core model...
```

Till now an initial model has been trained and necessary files are made by following these steps.

These screenshots show the initial training process. After this, we can talk to the initial bot.

We have only installed rasa x till this step, trained the initial model and completed installation.


```

Anaconda Prompt (anaconda3) - conda install ujson - conda install tensorflow
Finished creating project structure.
? Do you want to train an initial model? [Y] [n] Yes
Training an initial model...
Training Core model...
Processed Story Blocks: 100% | 5/5 [00:00:00:00, 2498.99it/s, # tr
Processed Story Blocks: 100% | 5/5 [00:00:00:00, 832.83it/s, # tr
Processed Story Blocks: 100% | 5/5 [00:00:00:00, 312.44it/s, # tra
Processed Story Blocks: 100% | 5/5 [00:00:00:00, 208.32it/s, # tra
Processed trackers: 100% | 5/5 [00:00:00:00, 625.36it/s, # ac
Processed actions: 16it [00:00:00, 8006.31it/s, # examples=16]
Processed trackers: 100% | 231/231 [00:01:00:00, 130.08it/s, # act
Epochs: 100% | 100/100 [00:12:00:00, 8.12it/s, t_loss=0.081, loss=0.008, a
2020-06-28 00:40:00 rasa.utils.tensorflow.models - Finished training.
2020-06-28 00:40:00 rasa.core.agent - Persisted model to 'C:\Users\aparn\AppData\Local\Temp\tmptxo4yqlo\core'
Core model training completed.
Training NLU model...
2020-06-28 00:40:00 rasa.nlu.training_data.training_data - Training data stats:
2020-06-28 00:40:00 rasa.nlu.training_data.training_data - Number of intent examples: 43 (7 distinct intents)
2020-06-28 00:40:00 rasa.nlu.training_data.training_data - Found intents: 'mood_great', 'goodbye', 'mood_unhappy', 'deny', 'bot_challenge', 'affirm', 'greet'
2020-06-28 00:40:00 rasa.nlu.training_data.training_data - Number of response examples: 0 (0 distinct responses)
2020-06-28 00:40:00 rasa.nlu.training_data.training_data - Number of entity examples: 0 (0 distinct entities)
2020-06-28 00:40:00 rasa.nlu.model - Starting to train component WhitespaceTokenizer
2020-06-28 00:40:00 rasa.nlu.model - Finished training component.
2020-06-28 00:40:00 rasa.nlu.model - Starting to train component RegexFeaturizer
2020-06-28 00:40:00 rasa.nlu.model - Finished training component.
2020-06-28 00:40:00 rasa.nlu.model - Starting to train component LexicalSyntacticFeaturizer
2020-06-28 00:40:00 rasa.nlu.model - Finished training component.
2020-06-28 00:40:00 rasa.nlu.model - Starting to train component CountVectorsFeaturizer
2020-06-28 00:40:00 rasa.nlu.model - Finished training component.
2020-06-28 00:40:00 rasa.nlu.model - Starting to train component CountVectorsFeaturizer
2020-06-28 00:40:00 rasa.nlu.model - Finished training component.
2020-06-28 00:40:00 rasa.nlu.model - Starting to train component DIETClassifier
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: You specified 'DIET' to train entities, but no entities are present in the training data. Skip train
Epochs: 100% | 100/100 [00:13:00:00, 7.62it/s, t_loss=1.682, i_loss=0.303, i_a
2020-06-28 00:40:24 rasa.utils.tensorflow.models - Finished training.
2020-06-28 00:40:25 rasa.nlu.model - Finished training component.
2020-06-28 00:40:25 rasa.nlu.model - Starting to train component EntitySynonymMapper
2020-06-28 00:40:25 rasa.nlu.model - Finished training component.
2020-06-28 00:40:25 rasa.nlu.model - Starting to train component ResponseSelector
2020-06-28 00:40:25 rasa.nlu.selectors.response_selector - Retrieval intent parameter was left to its default value. This response selector will be trained on training examples combining all
intents.
2020-06-28 00:40:25 rasa.nlu.model - Finished training component.
2020-06-28 00:40:25 rasa.nlu.model - Successfully saved model into 'C:\Users\aparn\AppData\Local\Temp\tmptxo4yqlo\nlu'
NLU model training completed.
Your Rasa model is trained and saved at 'C:\Users\aparn\OneDrive\Documents\Project\Careerbot\models\20200628-003930.tar.gz'.
? Do you want to speak to the trained assistant on the command line? [Y] [n] Yes
2020-06-28 00:44:29 root - Connecting to channel 'cmdline' which was specified by the '--connector' argument. Any other channels will be ignored. To connect to all given channels, omit the '-
' argument.
2020-06-28 00:44:29 root - Starting Rasa server on http://localhost:5005

```

11. Now talk to the initial bot and after that write **/stop** to stop the server

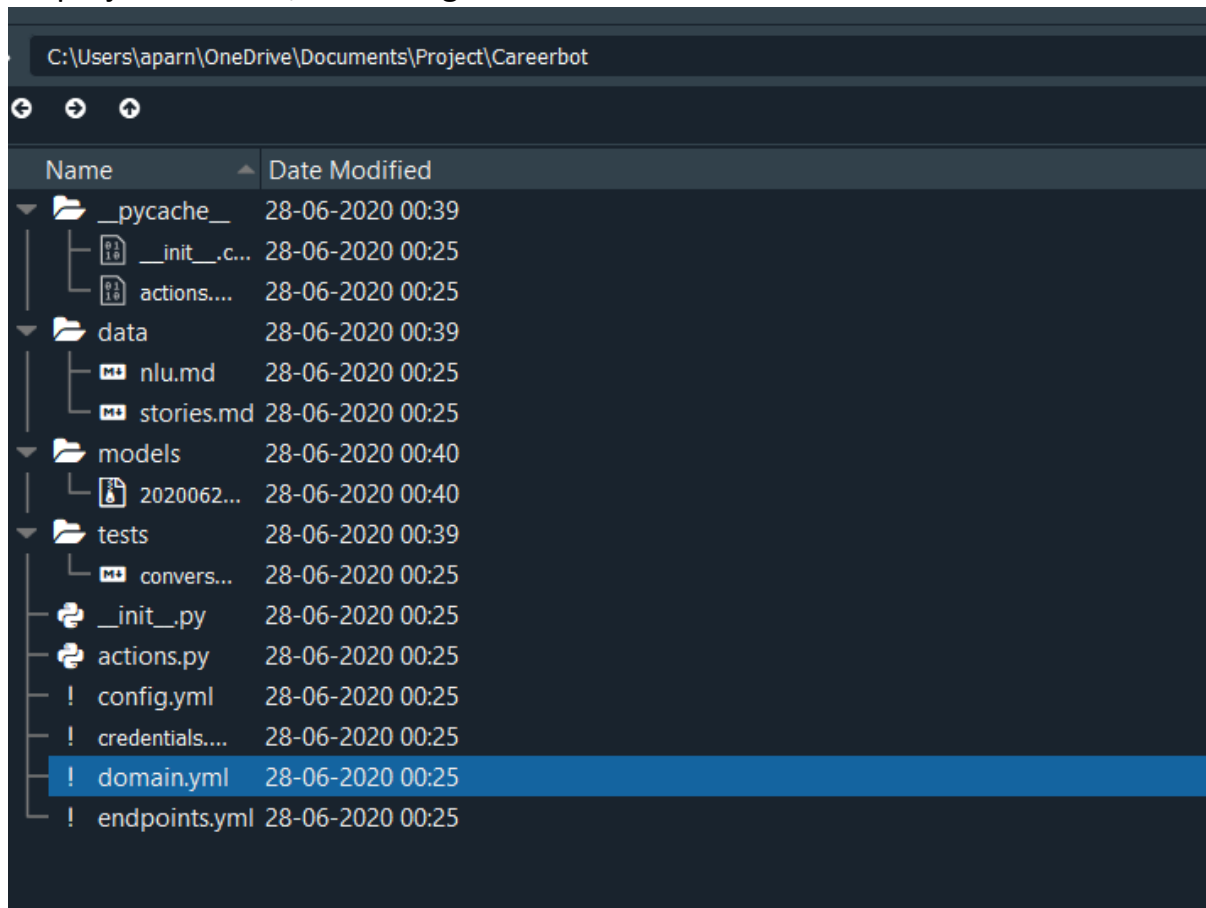
```

argument.
2020-06-28 00:44:29 root - Starting Rasa server on http://localhost:5005
2020-06-28 00:44:34 root - Rasa server is up and running.
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> hi
Hey! How are you?
Your input -> happy
Here is something to cheer you up:
Image: https://i.imgur.com/nGF1K8f.jpg
Did that help you?
Your input -> yep
Great, carry on!
Your input -> bye
Bye
Your input -> /stop
2020-06-28 00:49:16 root - Killing Sanic server now.

```

Files made by rasa init

Open Anaconda Navigator ,launch spyder and copy paste the path where project is saved, then we get some files which are shown below:



__init__.py

It is an empty file that helps python find your actions

Actions.py

It has code for your custom actions. In-case you want Rasa to call external server via REST API or API call, you can define your Custom Actions here.

config.yml

It contains configuration of your NLU and Core models.

Credentials.yml

Details for connecting to other services. In case you want to build Bot on Facebook Messenger, Microsoft Bot Framework, or Telegram you can maintain such credential and token here. I have discussed this in later sections.

I have integrated my chatbot using Telegram.

data/nlu.md

It contains NLU training data. Here you can define Intent.

We need to add related Sentences for that Intent.

data/stories.md

It contains your stories. This is required for Rasa Core. There is something called “Dialog Flow in Rasa” where Rasa Core controls the flow of the conversation between you and chatbot, so for that flow, you need to train chatbot using these stories.

domain.yml

It has assistant’s domain. This file combines Different Intent which chatbot can detect and list of Bot replies. Remember you can define your Custom Action Server Python method name here (in underscore format), so that Rasa will call that python method for you.

Endpoints.yml

For connecting to channels like FB messenger. This is mainly used for production setup.

models/<timestamp>.tar.gz

It contains the initial model

CHOOSING A PIPELINE

In Rasa Open Source, incoming messages are processed by a sequence of components. These components are executed one after another in a so-called processing pipeline defined in your config.yml. Choosing an NLU pipeline allows you to customize your model and finetune it on your dataset.

```
language: "en"
```

```
pipeline:
```

- **name:** ConveRTTokenizer
- **name:** ConveRTFeaturizer
- **name:** RegexFeaturizer
- **name:** LexicalSyntacticFeaturizer
- **name:** CountVectorsFeaturizer
- **name:** CountVectorsFeaturizer
- analyzer:** "char_wb"
- min_ngram:** 1
- max_ngram:** 4
- **name:** DIETClassifier
- epochs:** 100
- **name:** EntitySynonymMapper
- **name:** ResponseSelector
- epochs:** 100

IMPLEMENTATION

1. Open Spyder and make changes in 3 files namely **nlu.md**, **domain.yml** and **stories.md**
2. In nlu.md, add examples for each intent

I have made 7 intents on my own namely:

greetings,job_search,career_options,affirm,reject,feedback and exam.

have added atleast **150 training examples** for each of these to develop my chatbot's vocabulary.

We can remove the unnecessary intents that are made during initial training and can add as many intents and examples that are needed for us.

So, I kept an intent for **goodbye** , removed others and added these 7 intents.

Here is the description of each intent:

- **greetings** : for greeting the user like hi,hello,good morning etc.
- **job_search**: finds jobs for the user and addresses one's job queries
- **career_options**: for user's queries regarding what career to pursue
- **affirm**: to find out whenever user says yes or any other affirmation
- **reject**: to find out if the user says no
- **feedback**: for receiving user's feedback that can help in improvement of the bot
- **Exam**: addresses user's doubts regarding exams

A glimpse of some of my training examples are shown here:

```
## intent: exam
- which exam to appear for [medical](stream) ?
- is [gate](exam) necessary for [engineering](stream) students ?
- do [btech](course) students need to appear for [gate](exam) ?
- is it mandatory/ compulsory for [btech](course) students to take [gate](exam) exam ?
- essential exam for [btech](course) student
- xms for [btech student](course)
- exams for [btech](course) student
- compulsory xams after doing [btech](course)
- i am a [btech](course) student, after this where should i go for and what is the paper i need
- paper a [btech](course) student needs to clear to become [engg](position)
- what pprs do [btech](course) student should clr for job in [psu](company)
- essential xam for [btech](course) student
- if i want a job in [psu](company) , will i need to clear [gate](exam) or not ?
- i am interested in buliding my career in the field of [ai,ml](skill) ,what exams should i take
- do we need to clr any ppr for getting admission in [mtech](course)
- can we take admission in [mtech](course) without appearing for [gate](exam)
- is admission to [mtech](course) possible without [gate](exam)
- is service in [company_name](company) possible without clearing [gate](exam)
- can i get admitted to a good college in [mtech](course) without appearing for the entrance exam
- is there any alternative to [gate](exam)
- exams related to [gate](exam)
- how long does it take to prepare for [gate](exam) exam
- are [3 months](amount) sufficient to prepare for [gate](exam)
- can i get job after [btech](course) without [gate](exam)
- is a duration of [3 months](amount) ok to prepare for [gate](exam)
```

```
151 ## intent:affirm
152 - yes
153 - indeed
154 - of course
155 - that sounds good
156 - correct
157 - yes
158 - you can go further
159 - ok go ahead
160 - ok fine
161 - I agree
162 - agreed
163 - go ahead
164 - yes, fine
165 - ya
166 - haanji
167 - hnji
168 - haan
169 - you are right
170 - u r right
171 - okay move on
172 - yes,go ahead
173 - you are correct
174 - you're correct
175 - yep
176 - yup
177 - yaaaaas
178 - totally
```

```
## intent: feedback
- nice to talk to you
- good for nothing
- nice conversation
- really satisfied
- satisfied with your responses
- something i would really appreciate is that
- queries
- doubts
- i am pleased to talk to u
- extremely satisfied
- good replies
- i have some doubts regarding this
- really happy with replies
- fortunately got all answers
- answered all my questions
- lovely meeting
- would also like to meet you again
- will love to talk to u again
- extremely useful
- useful
- what i liked the most is that this chatbot gives very accurate replies
- thanks for giving me time
- glad to meet you
```

Similarly, add examples to other intents as well.

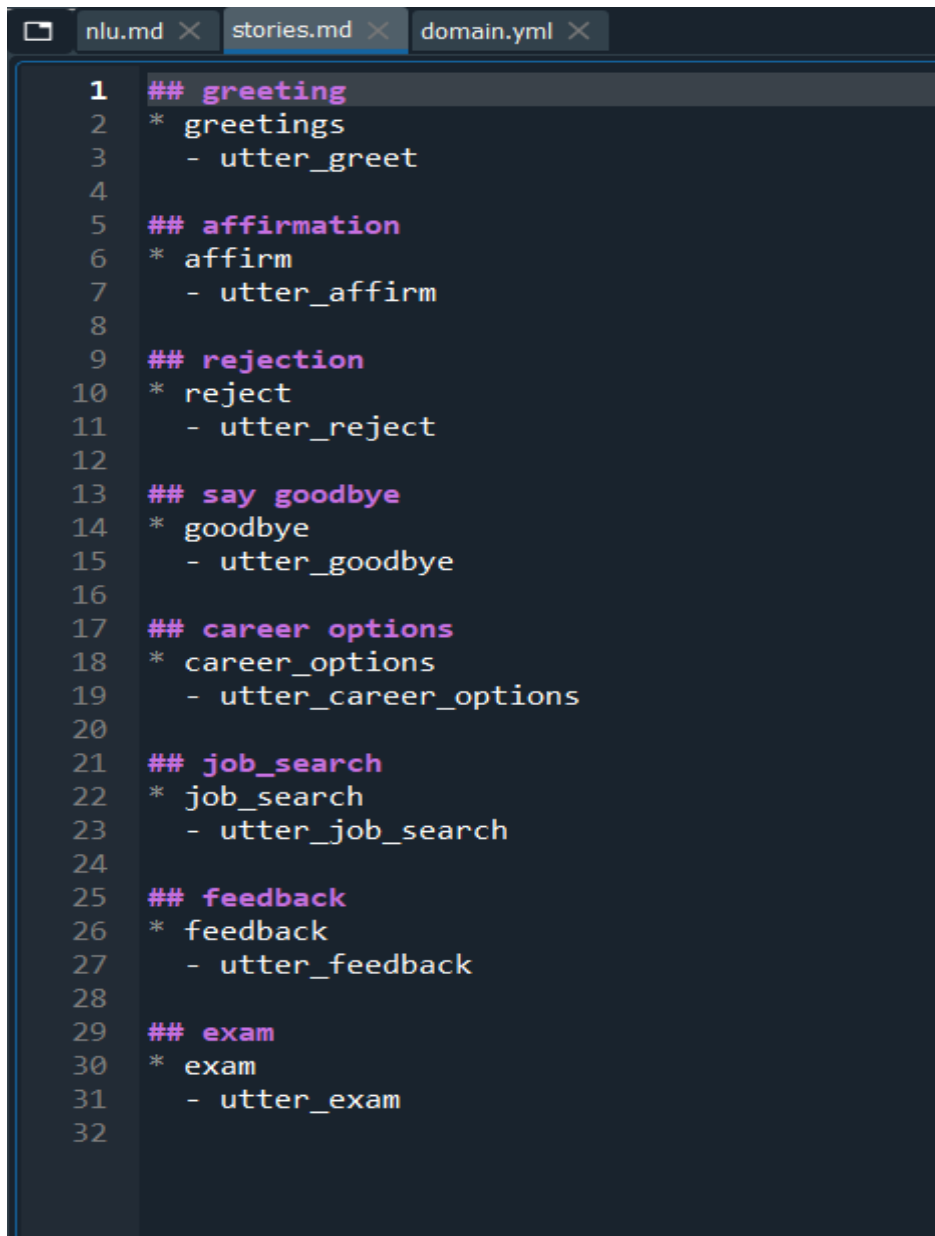
3. Open the 'domain.yml' file and put the following content:

```
- text: Sorry ask me another one
utter_feedback:
```

Add all intents, responses and actions in nlu.md file

4. Open the 'stories.md' file from the data folder and put the following content:

Stories.md file contains the sequence of dialog flow.



```
1  ## greeting
2  * greetings
3    - utter_greet
4
5  ## affirmation
6  * affirm
7    - utter_affirm
8
9  ## rejection
10 * reject
11   - utter_reject
12
13 ## say goodbye
14 * goodbye
15   - utter_goodbye
16
17 ## career options
18 * career_options
19   - utter_career_options
20
21 ## job_search
22 * job_search
23   - utter_job_search
24
25 ## feedback
26 * feedback
27   - utter_feedback
28
29 ## exam
30 * exam
31   - utter_exam
32
```

From the snapshot we can see that whenever chatbot identifies an intent for example when an intent greetings is found then we have written utter_greet for that and utter_greet 's action and responses are specified in domain.yml file.

What to respond is mentioned in domain.yml file whereas when to say what is specified in stories.md.

5. After all this, you can just enter the command **‘rasa train’** to train the model with new conversation elements.

```
(installingrasa) C:\Users\aparn\OneDrive\Documents\Project\Careerbot>rasa train
2020-06-29 04:16:16      rasa.model - Data (messages) for NLU model section changed.
Core stories/configuration did not change. No need to retrain Core model.
Training NLU model...
2020-06-29 04:16:17      rasa.nlu.training_data.training_data - Training data stats:
2020-06-29 04:16:17      rasa.nlu.training_data.training_data - Number of intent examples: 1164 (8 distinct intents)
2020-06-29 04:16:17      rasa.nlu.training_data.training_data - Found intents: 'exam', 'goodbye', 'reject', 'job_search', 'career_options', 'affirm', 'feedback', 'greetings'
2020-06-29 04:16:17      rasa.nlu.training_data.training_data - Number of response examples: 0 (0 distinct responses)
2020-06-29 04:16:17      rasa.nlu.training_data.training_data - Number of entity examples: 375 (29 distinct entities)
2020-06-29 04:16:17      rasa.nlu.training_data.training_data - Found entity types: 'name', 'position', 'college', 'qualification', 'suggestion', 'amount', 'role', 'subject', 'company', 'suggestion', 'location', 'branch', 'resource', 'location', 'class', 'qualification', 'quality', 'year', 'skill', 'stream', 'type', 'time', 'exam', 'position', 'better', 'course', 'category', 'board'
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'better' has only 1 training examples! The minimum is 2, because of this the training m
fail.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'board' has only 1 training examples! The minimum is 2, because of this the training m
fail.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'class' has only 1 training examples! The minimum is 2, because of this the training m
fail.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'field' has only 1 training examples! The minimum is 2, because of this the training m
fail.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'location' has only 1 training examples! The minimum is 2, because of this the traini
pay f
il.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'position' has only 1 training examples! The minimum is 2, because of this the traini
pay f
il.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'qualification' has only 1 training examples! The minimum is 2, because of this the tr
ng m
y fail.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'quality' has only 1 training examples! The minimum is 2, because of this the training
y fail.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'role' has only 1 training examples! The minimum is 2, because of this the training ma
il.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'suggestion' has only 1 training examples! The minimum is 2, because of this the train
may
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'suggestion' has only 1 training examples! The minimum is 2, because of this the trai
g may
fail.
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Entity entity 'type' has only 1 training examples! The minimum is 2, because of this the training ma
il.
2020-06-29 04:16:17      rasa.nlu.model - Starting to train component WhitespaceTokenizer
2020-06-29 04:16:17      rasa.nlu.model - Finished training component.
2020-06-29 04:16:17      rasa.nlu.model - Starting to train component RegexFeaturizer
2020-06-29 04:16:17      rasa.nlu.model - Finished training component.
2020-06-29 04:16:17      rasa.nlu.model - Starting to train component LexicalSyntacticFeaturizer
2020-06-29 04:16:18      rasa.nlu.model - Finished training component.
2020-06-29 04:16:18      rasa.nlu.model - Starting to train component CountVectorsFeaturizer
2020-06-29 04:16:19      rasa.nlu.model - Finished training component.
2020-06-29 04:16:19      rasa.nlu.model - Starting to train component CountVectorsFeaturizer
2020-06-29 04:16:20      rasa.nlu.model - Finished training component.
2020-06-29 04:16:20      rasa.nlu.model - Starting to train component DIETClassifier
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Mismatched entity annotation in message 'i am good in machine learning, which career should i pursue?
tent 'career_options'. Make sure the start and end values of entities in the training data match the token boundaries (e.g. entities don't include trailing whitespaces or punctuation).
More info at https://rasa.com/docs/rasa/nlu/training-data-format/
c:\users\aparn\anaconda3\envs\installingrasa\lib\site-packages\rasa\utils\common.py:363: UserWarning: Mismatched entity annotation in message 'do programmer needs to learn javascript' with intent 'career
tions
```

6. After the training is completed, enter the command **‘rasa x’** to test your chatbot in the web UI. You’ll see something like this :

The server is running at <http://localhost:5002/login?username=me&password=n8DDzwUAo9LL>

7. Copy this URL in your web browser and you’ll see the web UI for your chatbot:

8. Make the very first model as active because the very first model will be the recently trained one and then go to talk to your bot.

Talk to your bot (Interactive Learning)

utter_greet

Hey! I am a careerbot by Spotle. How can I help you?

action_listen

what i should do after doing my bsc

career_options{"course":"bsc"}

utter_career_options

We have received your query regarding career options and will send you an email shortly

when is neet exam in 2021

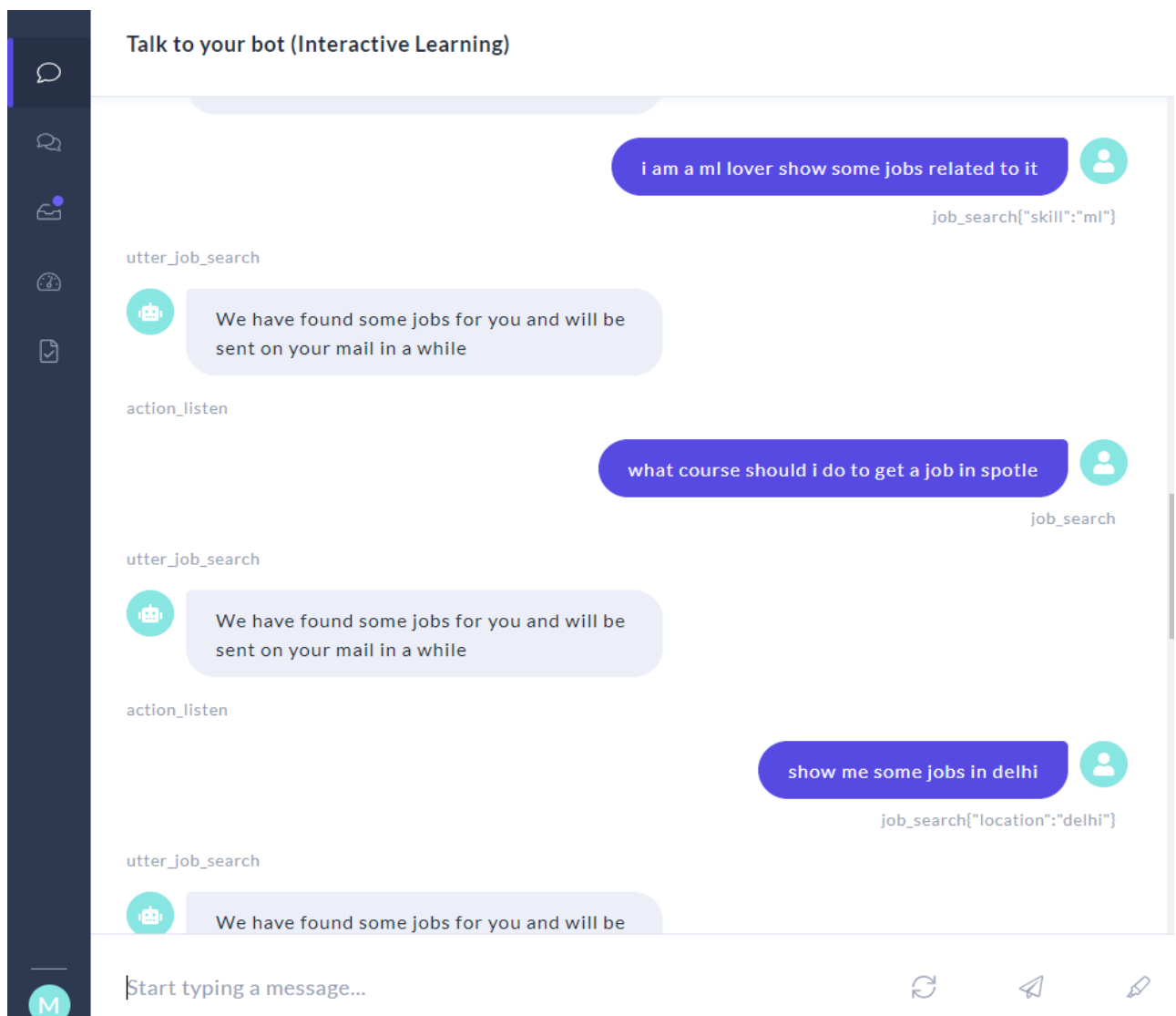
exam{"exam":"neet","year":"2021"}

utter_exam

We are trying our best to respond you with the exam details .Please check your mail

Start typing a message...

Test the chatbot by talking with it and check whether it is able to find out correct intents or not.



From these snapshots , we can see that chatbot is able to identify all intents as well as entities correctly.

It is identifying 2021 as 'year ' entity , bsc as 'course' , delhi as 'location' etc. All of which are correct.







TELEGRAM INTEGRATION

- Although we have tested our chatbot on the browser but the chatbot resides only on our system and if we want to publish it then we need to integrate it so that anyone could access my chatbot.
- I have used telegram for integration.

Steps :

1. Download ngrok from <https://ngrok.com/download>
2. After extracting the zip file, open the ngrok file and run it.
3. In ngrok, enter the command '**ngrok http 5005**' or '**ngrok.exe http 5005**'
4. It shows a url , copy it.
5. Then go to telegram and create your own bot using Botfather.
6. Open the telegram app and search for botfather(it is an inbuilt bot used to create other bots).
7. Start a conversation with botfather and enter `/newbot` to create a newbot.
8. Give a name to your bot
9. Give a username to your bot, which must end in `_bot`.This generates an access token.

BotFather bot

-  **Aparna** 10:44:14 AM
/newbot
-  **BotFather** 10:44:15 AM
Alright, a new bot. How are we going to call it? Please choose a name for your bot.
-  **Aparna** 10:44:43 AM
Careerbot
-  **BotFather** 10:44:44 AM
Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.
-  **Aparna** 10:45:14 AM
Aparna_Careerbot
-  **BotFather** 10:45:16 AM
Done! Congratulations on your new bot. You will find it at t.me/Aparna_Careerbot. You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

10. Open 'credentials.yml' and enter:

telegram:

access_token: "obtained from telegram"

verify: "your bot username"

webhook_url: "<https://<ngrokurl>/webhooks/telegram/webhook>"

<ngrokurl> is the url that was obtained by ngrok

11.Go to terminal and enter the command 'rasa run'

Anaconda Prompt (anaconda3) - rasa run

```
(rasa) C:\Users\aparn\OneDrive\Documents\Projects\Chatbot>rasa run
2020-06-30 12:30:43      root - Starting Rasa server on http://localhost:5005
2020-06-30 12:30:56      root - Rasa server is up and running.
```

12.Open one more terminal and run the command 'rasa run actions'

```
(rasa) C:\Users\aparn\OneDrive\Documents\Projects\Chatbot>rasa run actions
2020-06-30 12:33:33      rasa_sdk.endpoint - Starting action endpoint server...
2020-06-30 12:33:33      rasa_sdk.endpoint - Action endpoint is up and running on http://localhost:
```

13.Now, you can chat with your bot from Telegram.

RESULT AND OUTCOMES

The screenshot displays a chat interface on a mobile device. At the top, a blue header bar contains the text "Careerbot bot" on the left, a magnifying glass icon for search on the right, and a three-dot menu icon on the far right. The chat messages are organized into a list. Each message is preceded by a circular avatar: a red circle with "AP" for the user and a yellow circle with "CA" for the bot. The messages are as follows:

- AP:** "hi" (12:38:05 PM)
- CA:** "Hey! I am a careerbot by Spotle. How can I help you?" (12:38:07 PM)
- AP:** "show some career options for btech passout" (12:39:08 PM)
- CA:** "We have received your query regarding career options and will send you an email shortly" (12:39:10 PM)
- AP:** "when is jee exam" (12:40:08 PM)
- CA:** "We are trying our best to respond you with the exam details .Please check your mail after a while" (12:40:31 PM)
- AP:** "i am skilled in ai show some jobs related to this" (12:41:48 PM)
- CA:** "We have found some jobs for you and will be sent on your mail in a while" (12:41:54 PM)

At the bottom of the screen, there is a white input area. On the left is a red circle with "AP". In the center is a text input field with the placeholder text "Write a message...". To the right of the input field is a yellow circle with "CA" and a smiley face icon. Below the input field is a horizontal line, and at the very bottom is a row of icons: a document, a camera, a microphone, a crying face, a heart, a smiling face with hearts, a smiling face, a grimacing face, and a thumbs up. To the right of these icons is a blue button labeled "SEND".

Careerbot bot



AP

Aparna

12:45:03 PM

is there any job suitable for bsc passout

CA

Careerbot

12:45:11 PM

We have found some jobs for you and will be sent on your mail in a while

AP

Aparna

12:48:00 PM

what to do after completing bcom

CA

Careerbot

12:48:05 PM

We have received your query regarding career options and will send you an email shortly

AP

Aparna

12:48:53 PM

give me syllabus for gate 2021

CA

Careerbot

12:49:01 PM

We are trying our best to respond you with the exam details .Please check your mail after a while

AP

Aparna

12:51:43 PM

is jee compulsory to get admission in btech

CA

Careerbot

12:52:10 PM

We are trying our best to respond you with the exam details .Please check your mail after a while

AP

Write a message...



CA



SEND

Careerbot bot



Aparna
namashkar

12:59:59 PM



Careerbot
Hey! I am a careerbot by Spotle. How can I help you?

1:00:06 PM



Aparna
i am good at maths what course should i take

1:00:58 PM



Careerbot
We have received your query regarding career options and will send you an email shortly

1:01:05 PM



Aparna
what exams should i take for admission to dtu

1:04:56 PM



Careerbot
We are trying our best to respond you with the exam details .Please check your mail after a while

1:05:01 PM



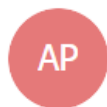
Aparna
i am not happy

1:06:10 PM



Careerbot
Thanks for your valuable feedback.Nice to talk to you,goodbye

1:06:12 PM











Write a message...



SEND

Careerbot bot

-  **Aparna** 1:07:23 PM
i agree with you
-  **Careerbot** 1:07:32 PM
Great,carry on
-  **Aparna** 1:08:21 PM
good suggestions
-  **Careerbot** 1:08:23 PM
Thanks for your valuable feedback.Nice to talk to you,goodbye
-  **Aparna** 1:10:00 PM
nope
-  **Careerbot** 1:10:17 PM
Sorry,ask me another one
-  **Aparna** 1:10:46 PM
its definitely possible
-  **Careerbot** 1:11:15 PM
Great,carry on

CONCLUSION

I have successfully made a chatbot using Rasa X with a good accuracy and also integrated it with Telegram .

As the snap shots show clearly :

- Bot is able to identify each intent very well
- Bot also handles denial from user successfully
- Bot is able to identify whether the user affirms or denies.
- This Careerbot can answer queries related to career options, exams and job related.

REFERENCES :

1. Rasa Official documentation <https://rasa.com/docs/rasa/user-guide/installation/>
2. <https://rasa.com/docs/rasa/user-guide/architecture/>
3. <https://rasa.com/docs/rasa/user-guide/rasa-tutorial/>
4. <https://rasa.com/docs/getting-started/>
5. <https://rasa.com/docs/rasa-x/>