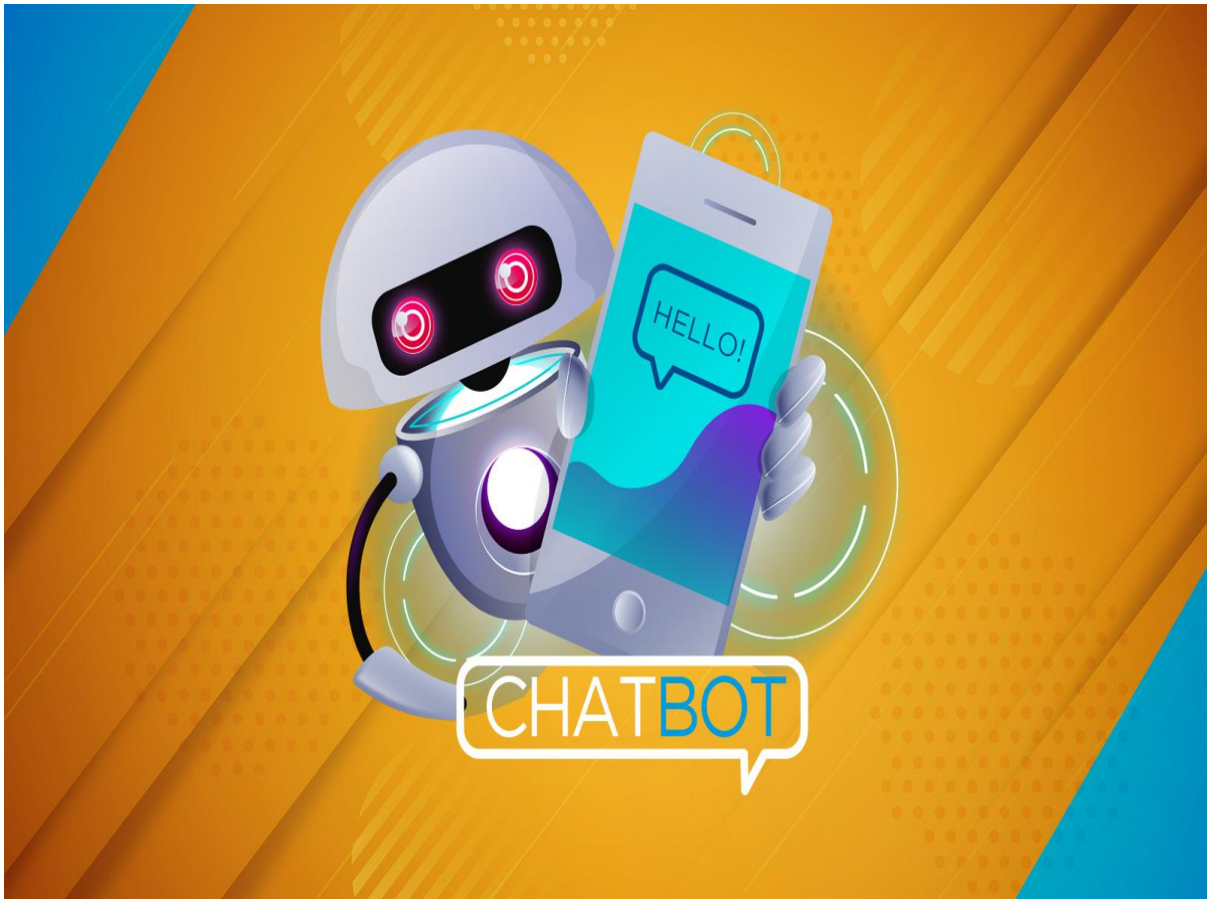
CAREERBOT USING RASA FRAMEWORK



BY- APARNA

EMAIL- aparnaaugust3@gmail.com

TABLE OF CONTENTS

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

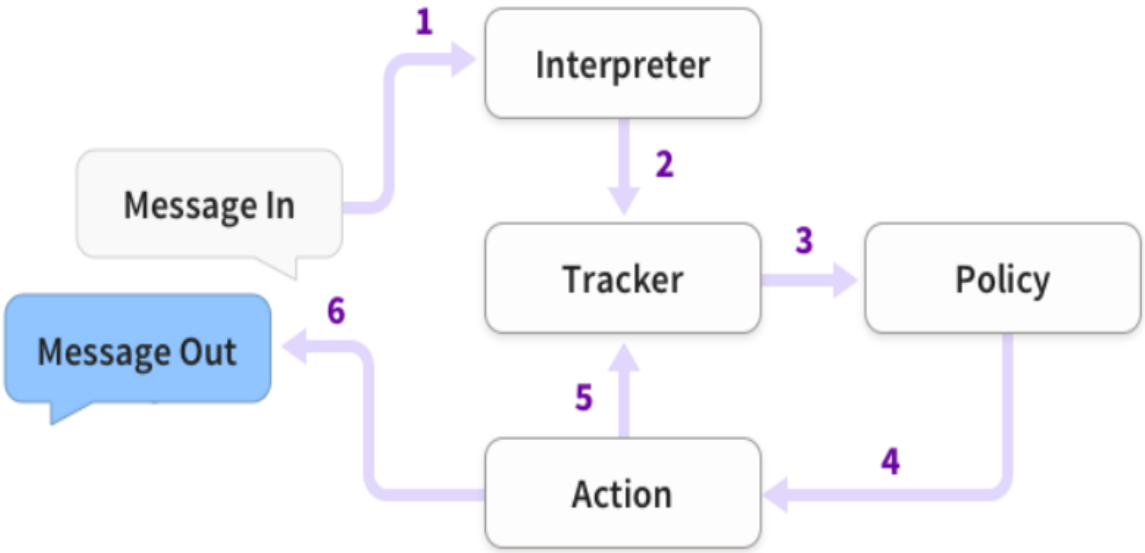
Rasa NLU

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

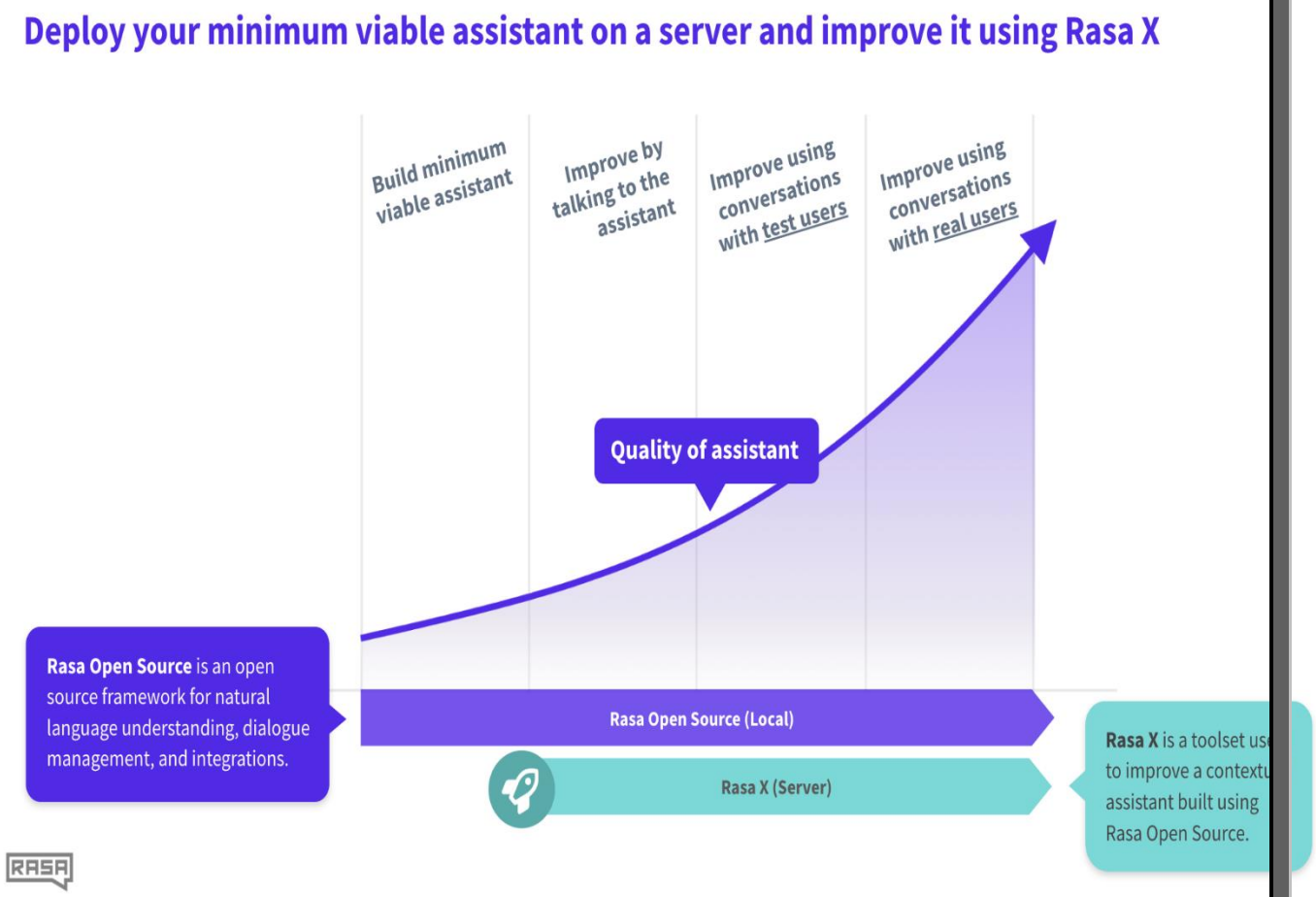
- **NLU**: determines what the user wants and captures key contextual information
- **Core:** selects the next best response or action based on conversation history
- **Channels and integrations:** connect assistant to users and backend systems

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



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++

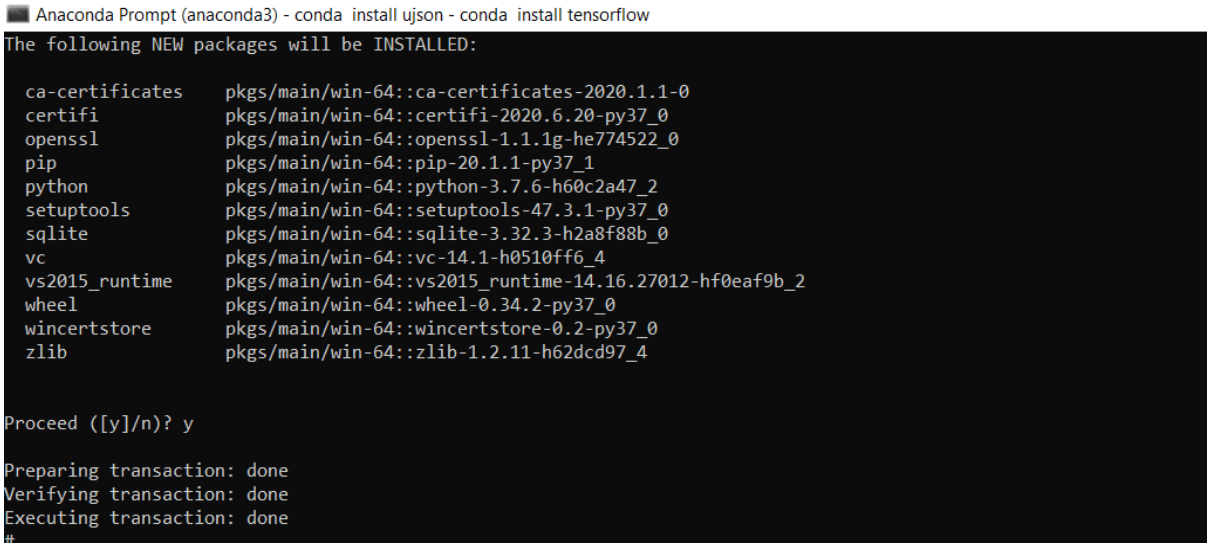
INSTALLATION OF RASA X

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



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.

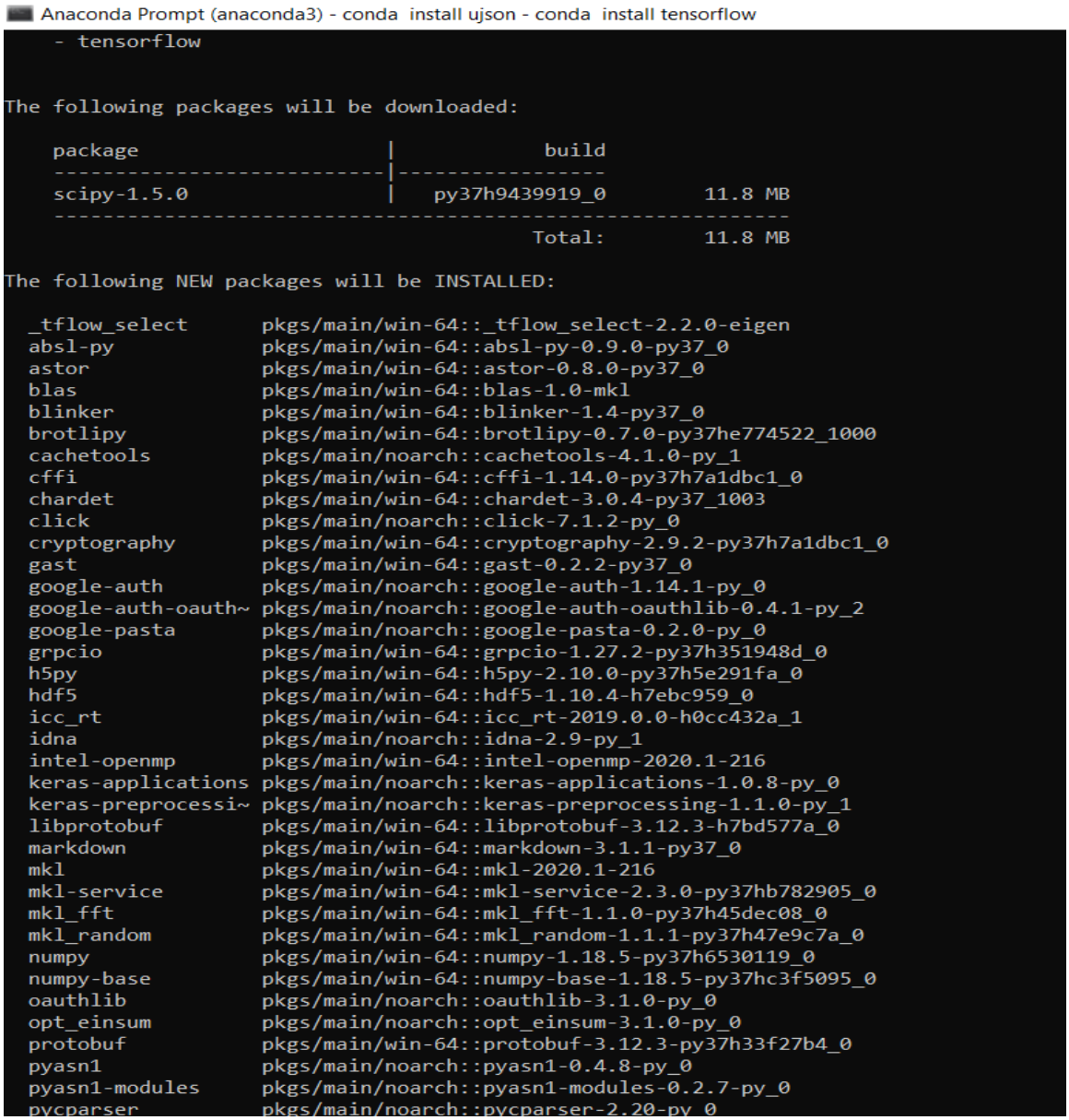
7. Then, **conda install tensorflow**

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:
```



8. Now install rasa x using following command:

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

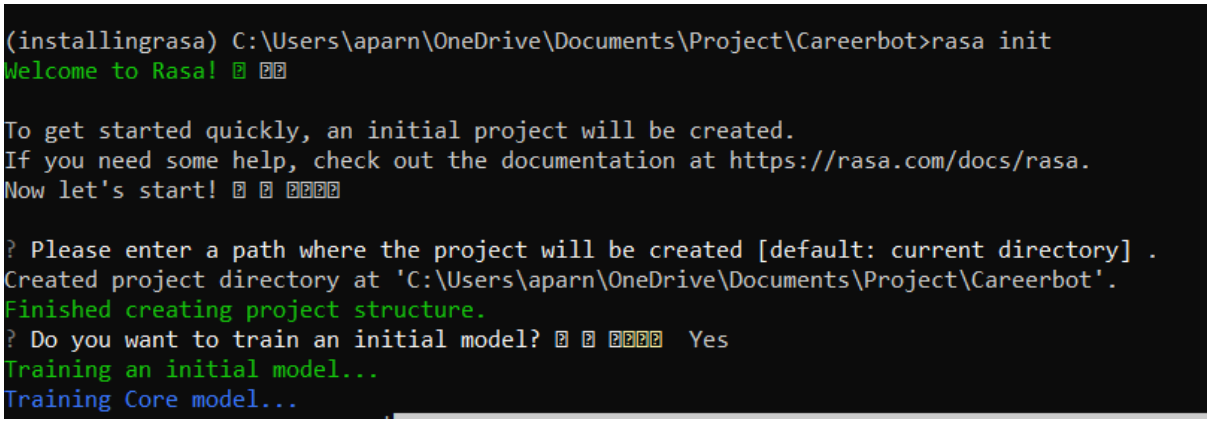
9. Now install spacy. **spaCy** is a free open-source library for Natural Language Processing in **Python**.

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.

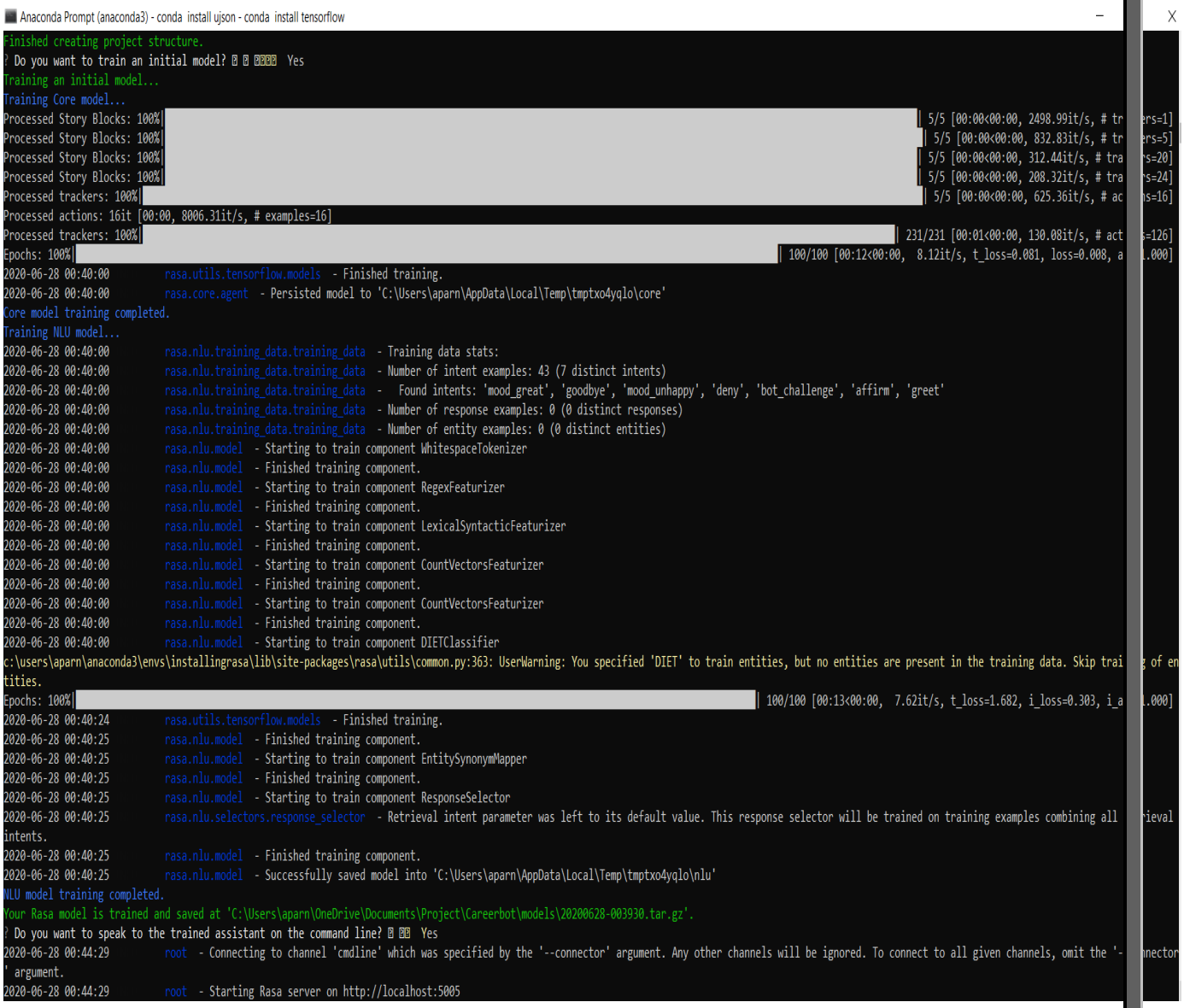
rasa init



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.

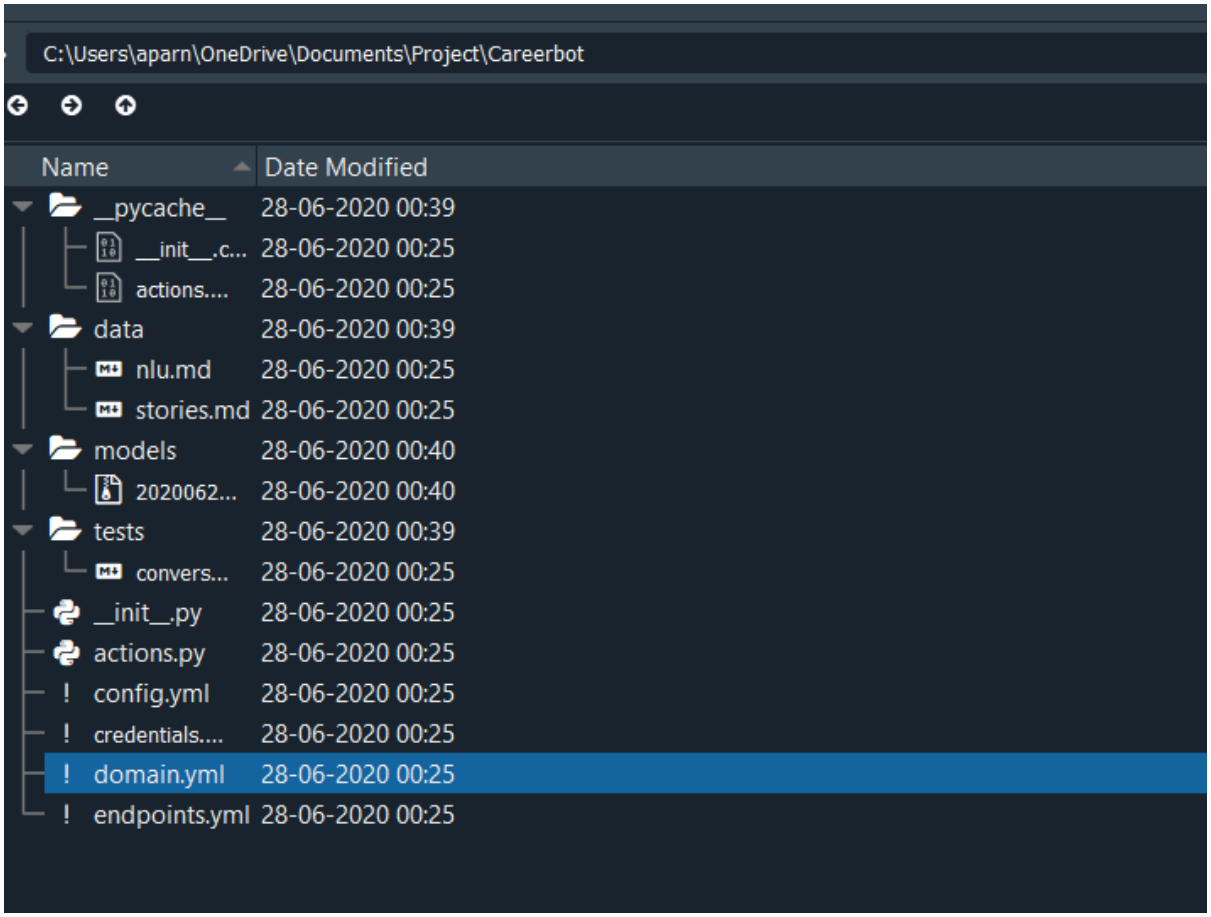


11. Now talk to the initial bot and after that write **/stop** to stop the server

```
  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.

```yaml
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. I have added atleast **150 training examples f**or each of these to develop my chatbot's vocabulary.

We can remove the unncessary 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 nee
- 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 ta
- 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 e
- 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 converation
- 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
```

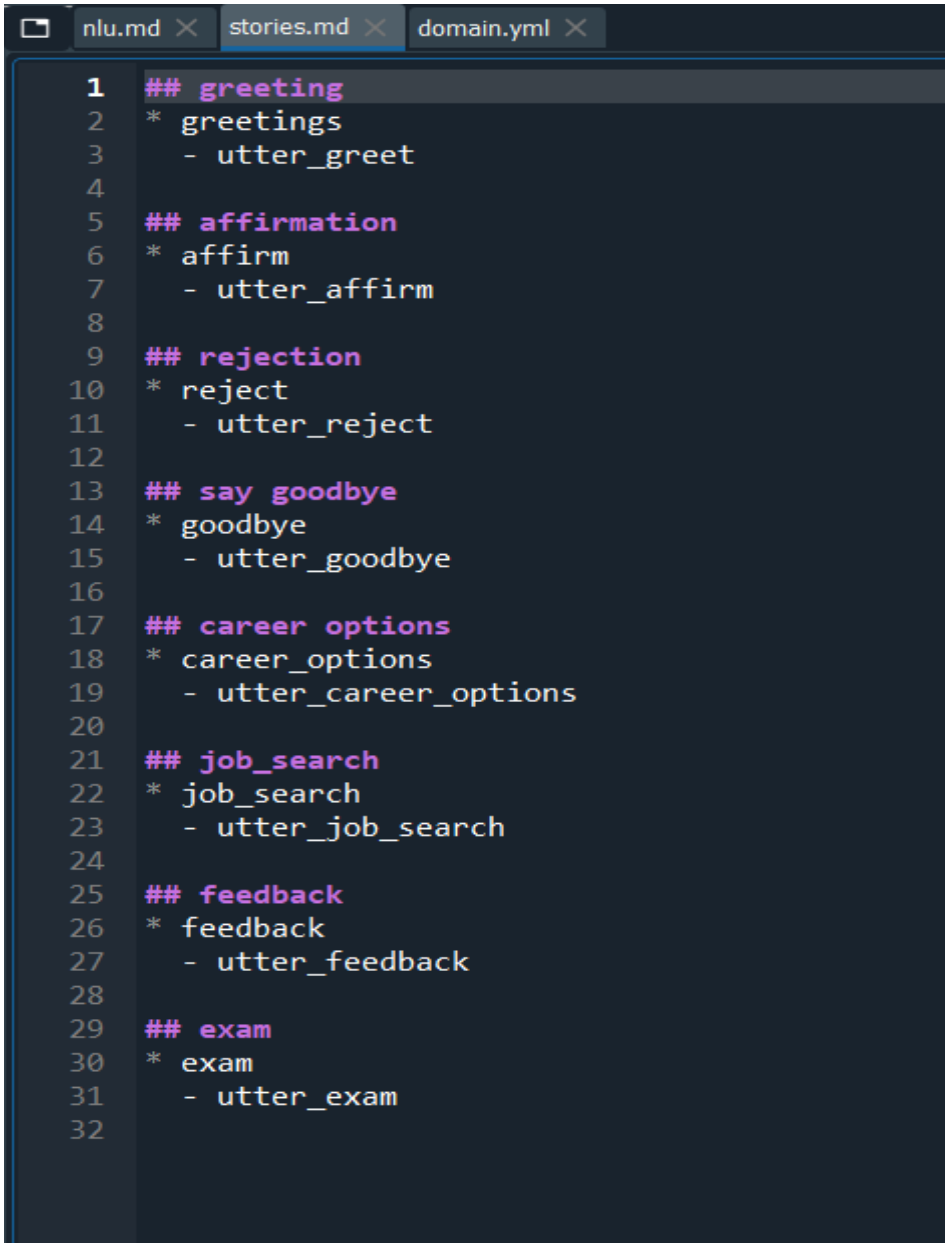
Similarly, add examples to other intents as well.

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

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.



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.

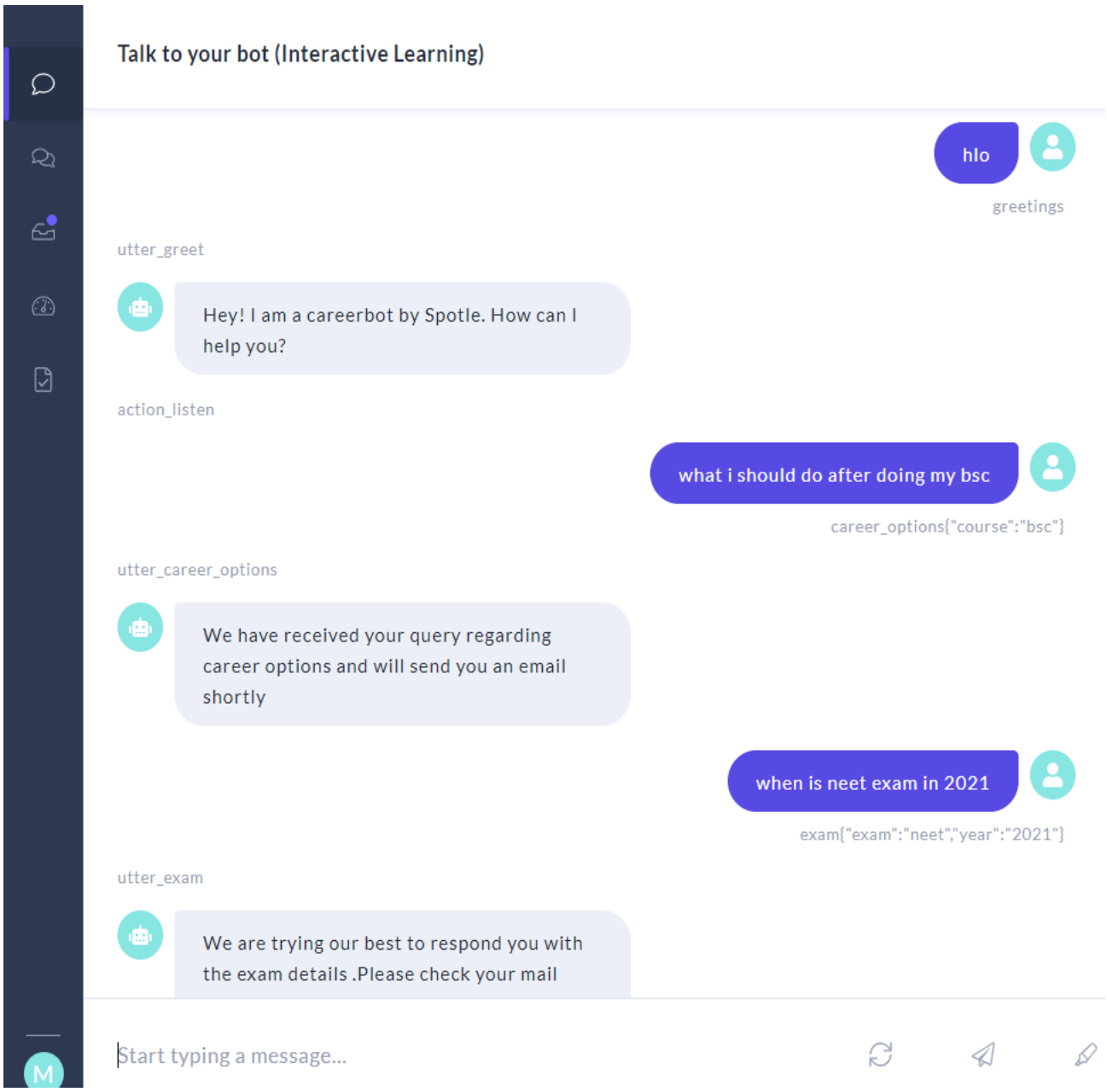
5. After all this, you can just enter the command '**rasa train'** to train the model with new conversation elements.

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 :

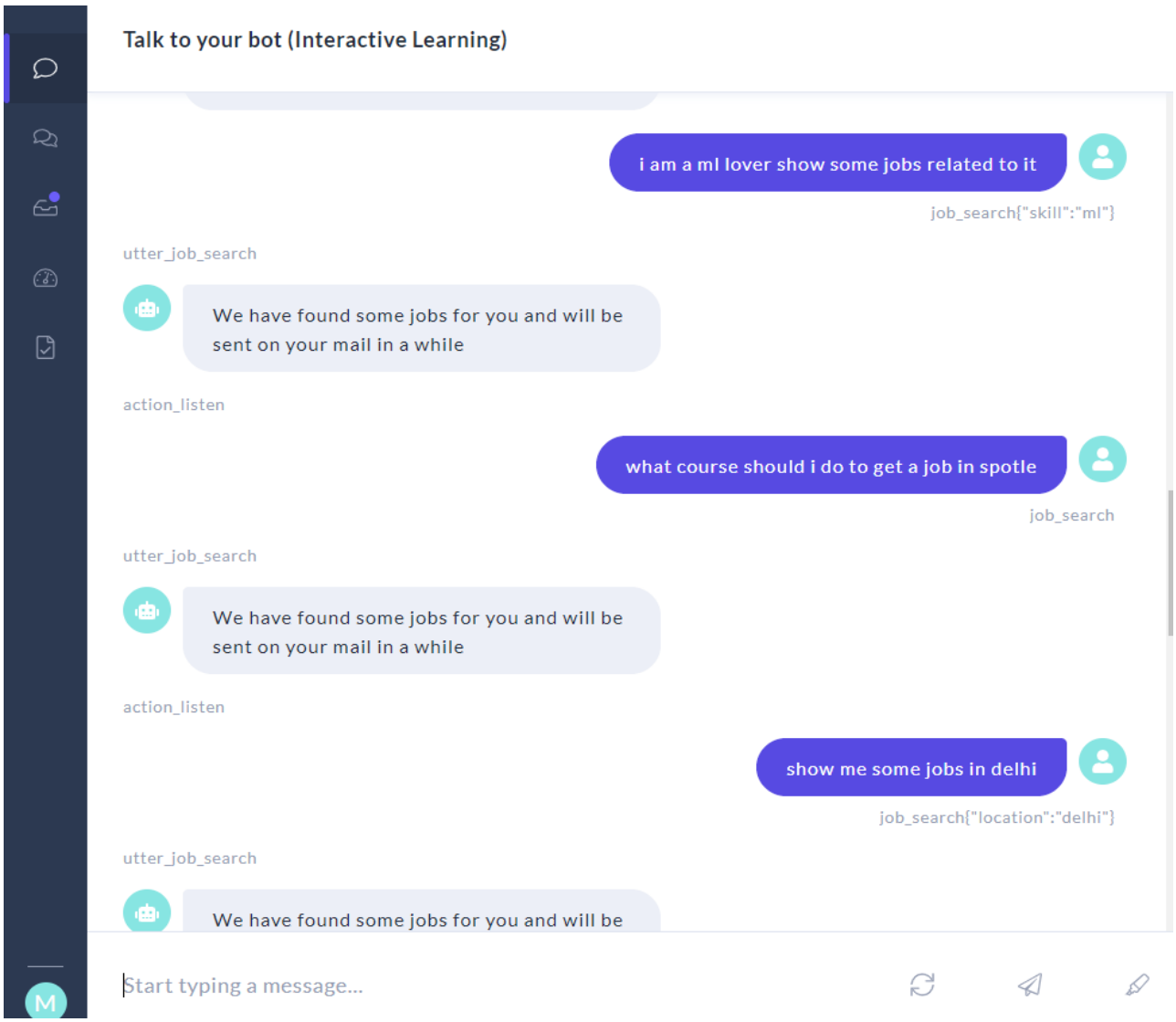
```
The server is running at http://localhost:5002/login?username=me&password=n8DDzwUAo9LL
```

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.



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.

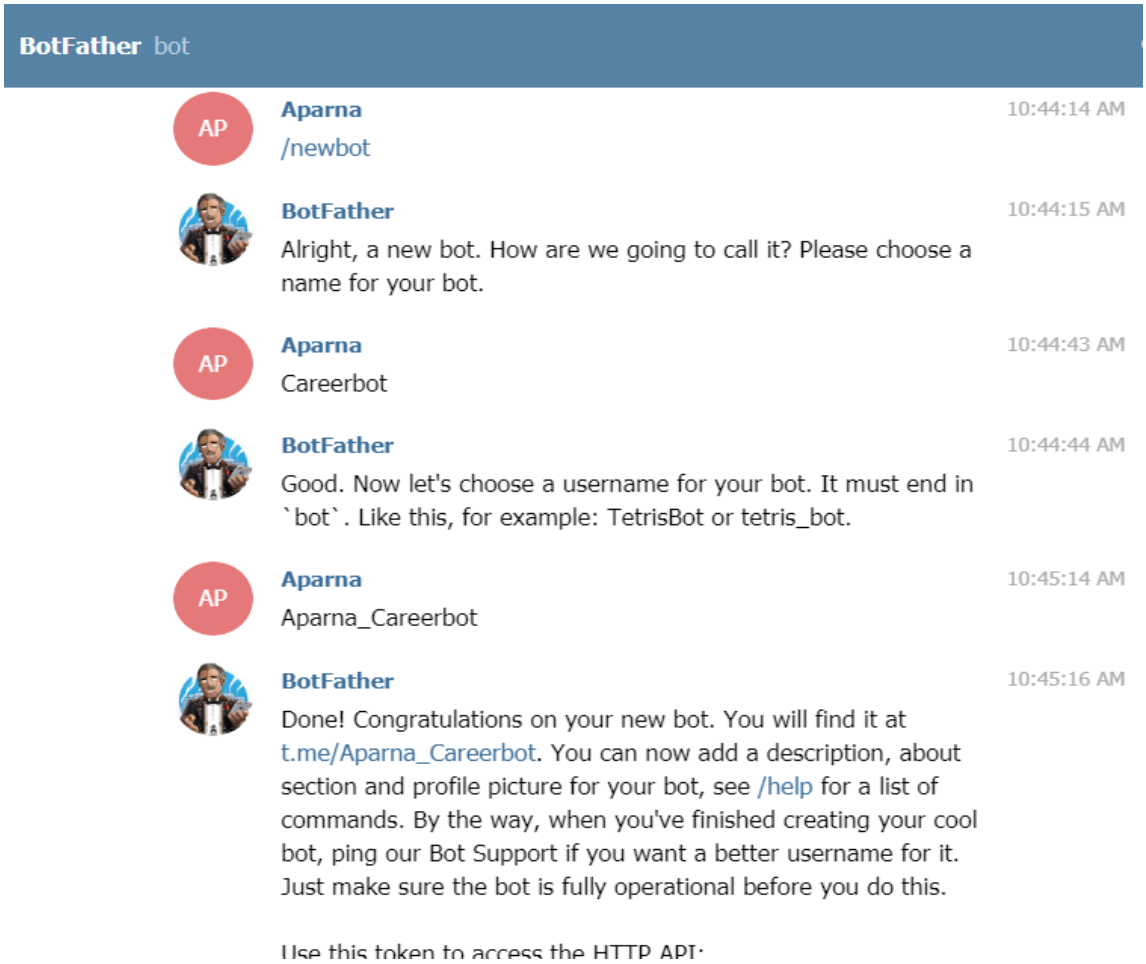
It is identifying 2021 as ‘year ‘ entity , bsc as ‘course’ , delhi as ‘location’ etc. All of which are correct.

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.



10.Open ‘credentials.yml’ and enter:

telegram:

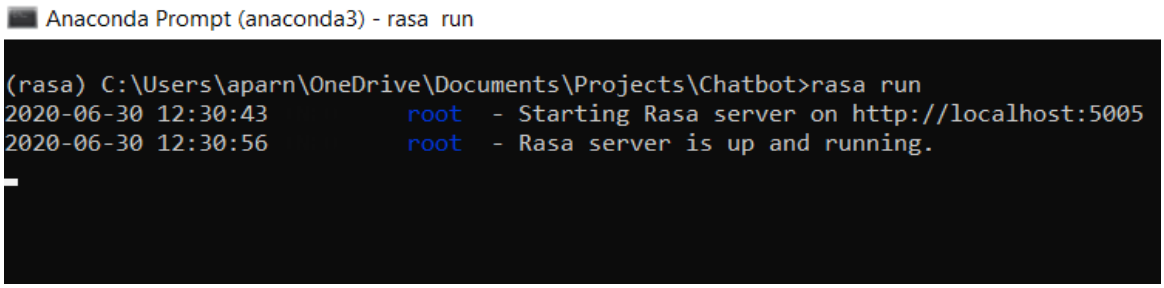
access_token: "obtained from telegram"

verify: "your bot username"

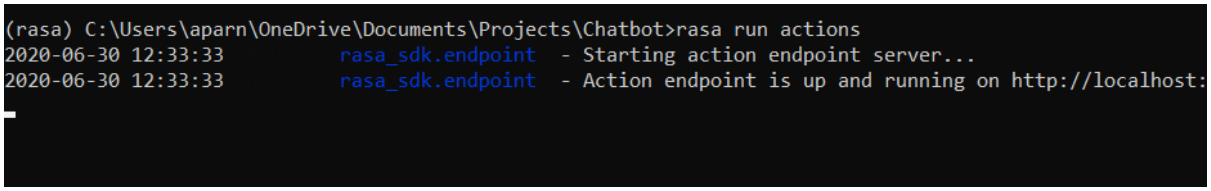
webhook_url: "https://<ngrokurl>/webhooks/telegram/webhook"

<ngrokurl> is the url that was obtained by ngrok

11.Go to terminal and enter the command ‘rasa run’

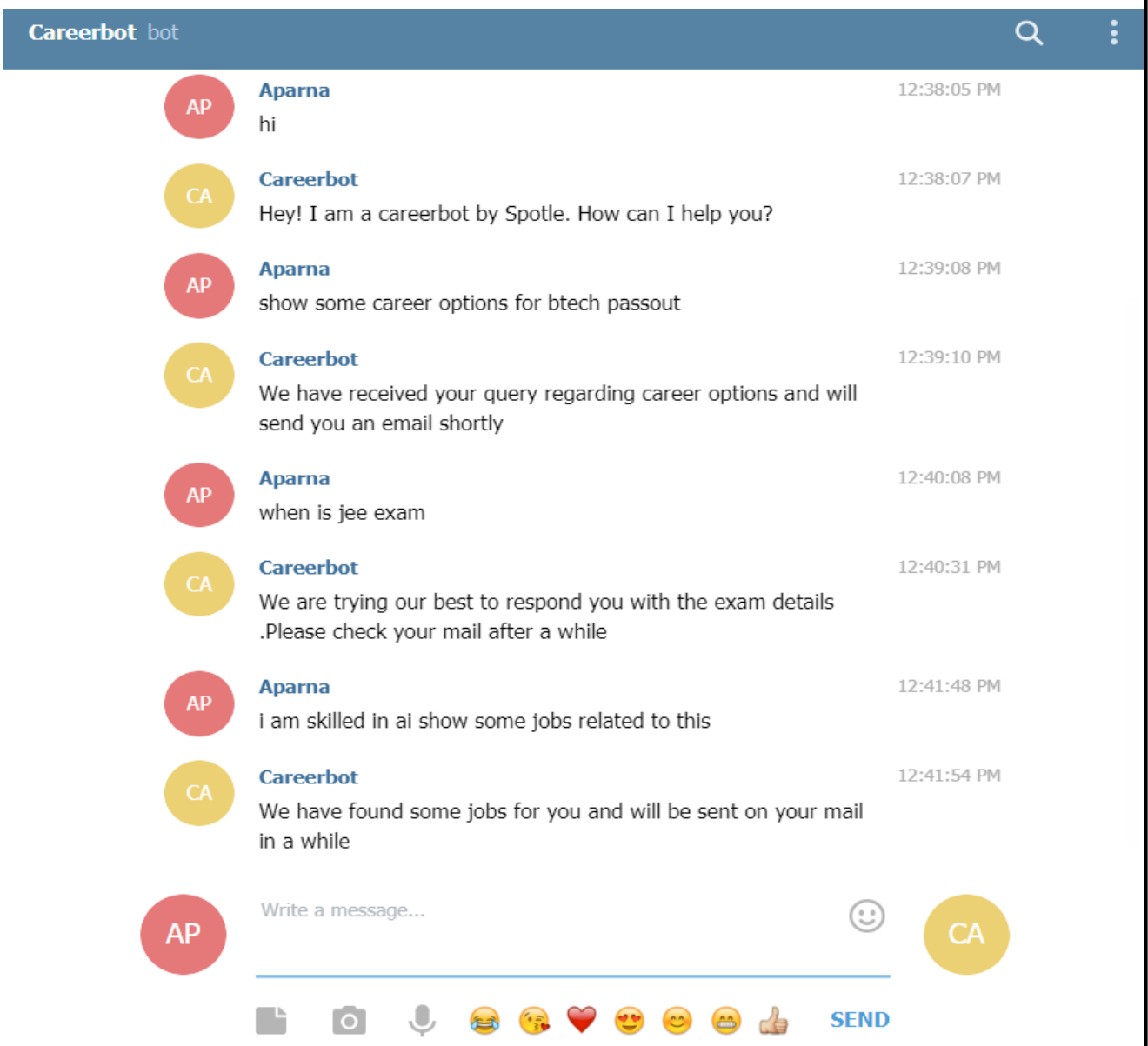


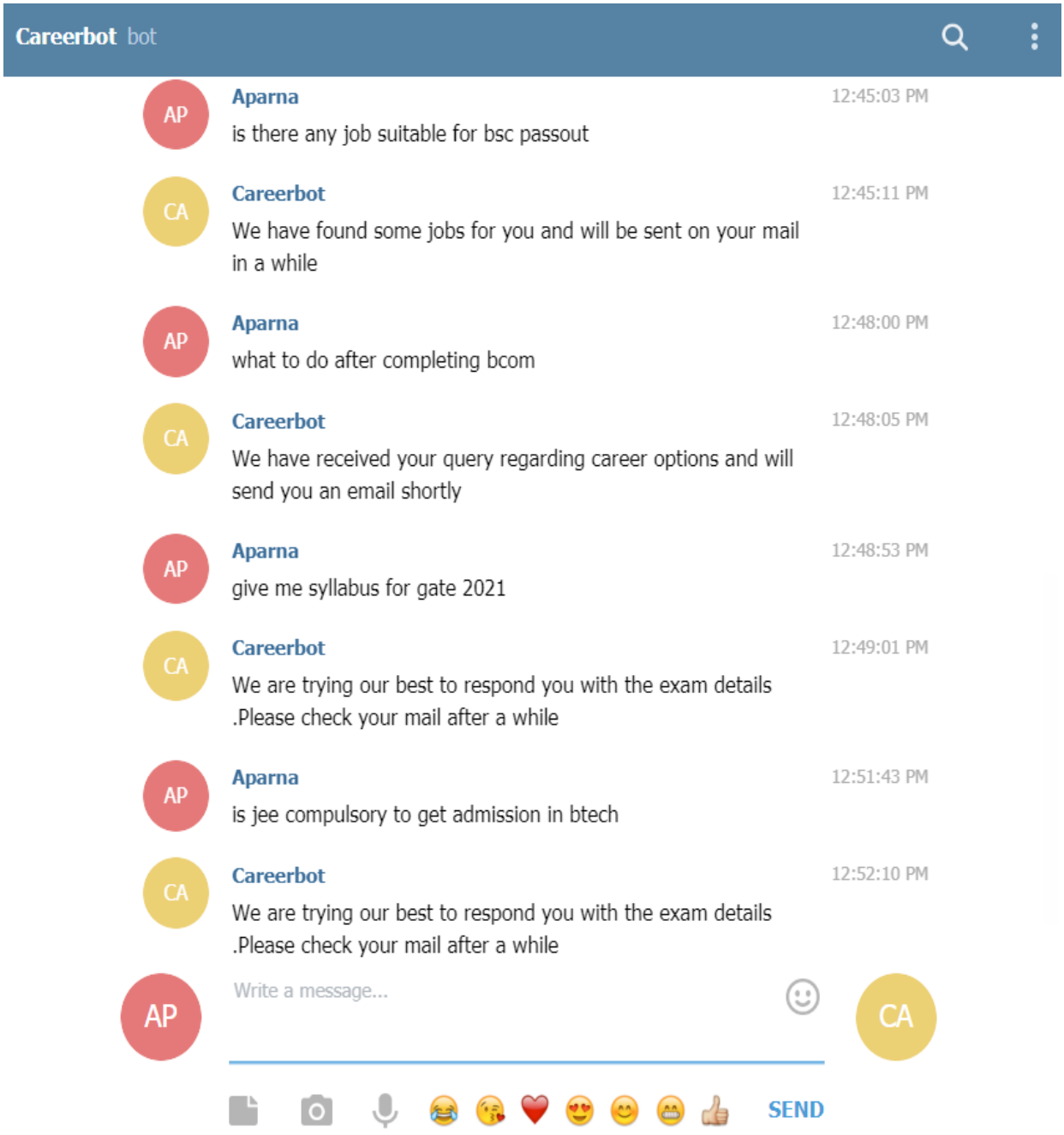
12.Open one more terminal and run the command ‘rasa run actions’

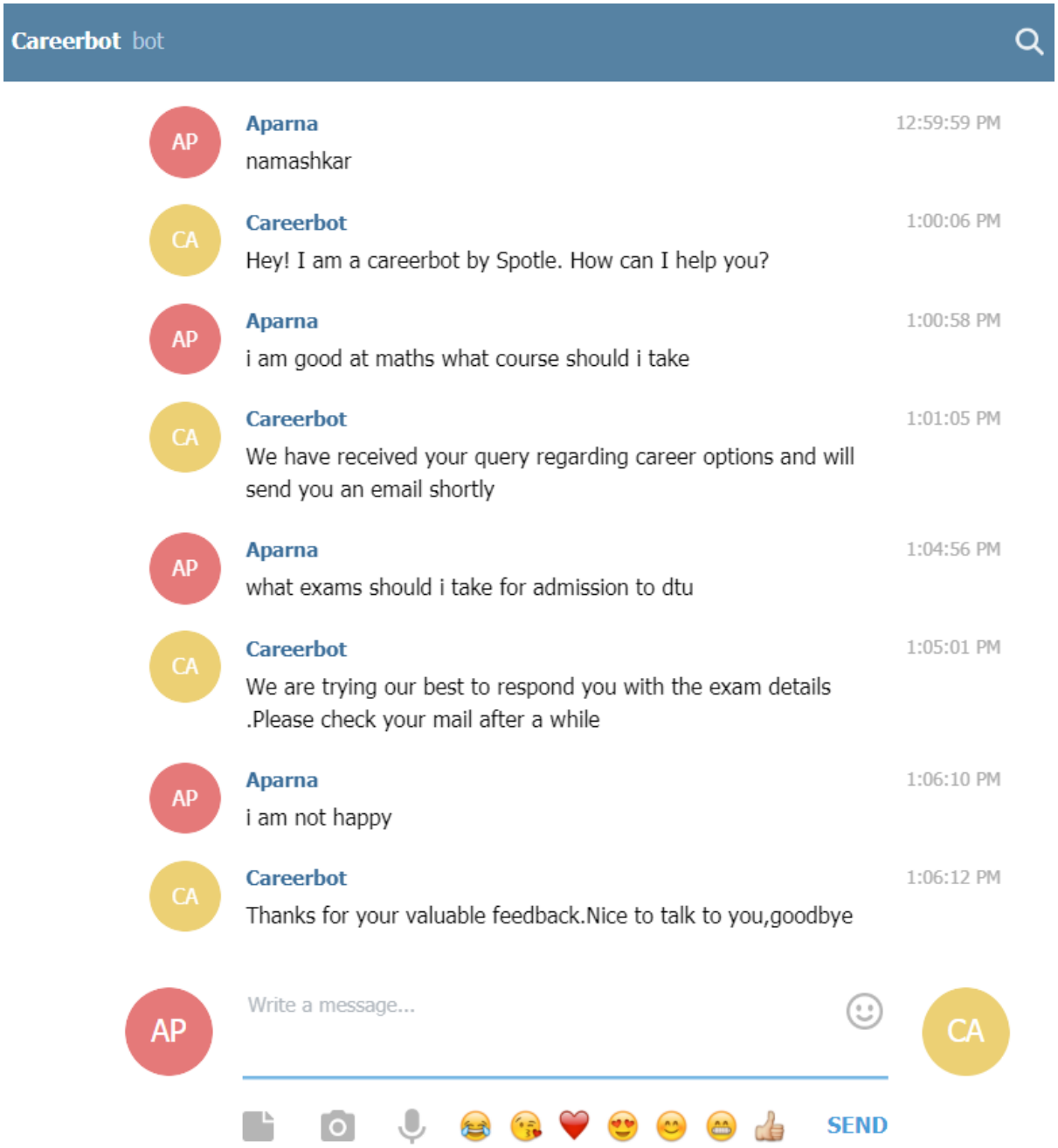


13.Now, you can chat with your bot from Telegram.

RESULT AND OUTCOMES




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
AP
Aparna
12:59:59 PM
namashkar
CA
Careerbot
1:00:06 PM
Hey! I am a careerbot by Spotle. How can I help you?
AP
Aparna
1:00:58 PM
i am good at maths what course should i take
CA
Careerbot
1:01:05 PM
We have received your query regarding career options and will send you an email shortly
AP
Aparna
1:04:56 PM
what exams should i take for admission to dtu
CA
Careerbot
1:05:01 PM
We are trying our best to respond you with the exam details .Please check your mail after a while
AP
Aparna
1:06:10 PM
i am not happy
CA
Careerbot
1:06:12 PM
Thanks for your valuable feedback.Nice to talk to you,goodbye
AP
Write a message...
CA
SEND

Careerbot bot

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