A close up of a flower

Description automatically generated

**Transport Chatbot Powered by RASA**

**Presented by- Nirvi Khiyani**

**Student ID :3901390**

**Module Code : CSI\_7\_FIT**

**Table of Contents**

1. **Abstract 3**
2. **Introduction 3**

**3. Chatbot definition, its importance and working of chatbot 3**

**3.1 Definition of chatbot 4**

**3.2 Types of Chatbot 5**

**3.3 Importance of Chatbots 5**

**3.4 Working of Chatbot 5**

**3.5 Purpose of Transport chatbot 6**

**4.Scope of the project and tasks done by me and my team 6**

**4.1 Task done by me 6**

**4.2 Task done by team 11**

**5. Framework 11**

**5.1 Components of Rasa 11**

**5.2 Architecture of RASA 13**

**5.3 Why Rasa over other platform 14**

**5. 4 Intent Classification Architecture 14**

**5.5 RASA x 14**

**5.6 Features provided by RASA: 14**

**5.7 Description of Chatbot pipeline 15**

**6. Coding and Implementation 15**

**6.1 Coding 15**

**6.2 Implementation 16**

**7. Execution 23**

**7.1 Execution part 23**

**7.2 Execution errors 28**

**8.Performance and Accuracy 30**

**9. Conclusion** 30

**10. References 31**

**1. Abstract:**

The popular issues for intelligent systems for the in the field of application are natural language processing and robot system. Due to development of neural network and natural language processing, artificial intelligence application is expanding the Chatbots systems, that are actually used for practical purposes which includes information acquisition and customer service. This report includes functional framework and gives introduction the principles of Chatbot RASA NLU, which integrate Rasa NLU as well as methods of neural network and implementation on entity after the recognition of user intent. This report highlights development of Transport chatbot, challenges faced while developing chat and how those challenges were overcome. This chatbot was developed by using Rasa, Pycharm and Python. This bot helps the user to book a cab using mobile number and postal code and the results demonstrate engagement of user and proving a complete answer in this domain

**2. Introduction:**

The theoretically advanced technique for Natural Language Processing is understanding human beings automatically and representing their language [1] and is the major area for AI that is used in text mining, speech recognition, machine translation. There are various NLP phases’ morphology, syntax, pragmatics, semantics and phonetics as well. For understanding human language, machine divides the entire task into sentences, phrases, words and paragraphs. On top of that, it should recognize association within words, get the exact meaning from text, and consider the sentences for various situations. A model for feed- forward neutral language was proposed in the early 21st century that uses word embedding along with implementation of word2vec that is efficient to get relation within words. Also, recently recurrent neural network is the replacement of feed-forward neural networks and long-term memory network for modelling of language [2]

As an NLP application, Chabot’s made it easier for computer for understanding user’s intent of natural language to give the required response and made it popular on various platforms. From different chatbots kinds, the major one is “virtual-assistants” that supports user’s requirements in various sectors and domain. The main parts of chatbots system are NLU part that states the intent of user, dialog management and conversation state and NLU part of Generation for responding user’s queries.

The important role in the whole system is NLU and there are many methods that help to understand and respond. To build controversational systems, RASA NLU is used that is an open-source natural language module. It consists of various models that are coupled and are combined with natural language processing and libraries of machine learning in API [3]. Rasa makes a prediction of slot labels and values that are related with various input segments ispite of slots of sequence for each word [4]. This report highlights the development of chatbot to book a cab, cancel a cab and handle with any of the other exceptional requests other than booking and cancelling of cab. Section 3 highlights definition, working and importance of chatbot. Section 4 states the framework which includes components, architecture of Rasa and pipeline description. Section 5 is for Coding, Implementation and the errors description while implementation. Section 6, which includes execution part which shows the execution and the errors while execution. And finally, section 7 concludes the report.

**3. Chatbot definition, its importance and working of chatbot:**

**3.1 Definition of chatbot:**

Chatbot is an Artificial Intelligence tool that can simulate a conversation in natural language via mobile apps, messaging applications, through telephone and website. Chatbots are virtual assistants which are used in almost all industries to simply the interaction between computers and human. Chatbot is a feature of Artificial Intelligence which is used and embedded by many different messaging apps. Some of the examples of AI chatbot are Siri, Google Assistants Amazon’s Alexa and messaging apps like Facebook messenger and WeChat.

**3.2 Types of Chatbot:**

Chatbots are built on the technologies of [artificial intelligence](https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence) (AI), which includes natural language processing, [machine learning](https://searchenterpriseai.techtarget.com/definition/machine-learning-ML) (ML) algorithms and deep learning, and require massive data. The more interaction of user with the bot, it will create better recognition of [voice](https://searchcustomerexperience.techtarget.com/definition/voice-recognition-speaker-recognition) and predicts proper response. There are various types of chatbot as stated under:

1. **Scripted or quick reply chatbots: These are basic chatbot that interacts with the user by a set of pre-defined questions that progresses till the chatbot has the answer. These types are like menu based where user must choose from pre-defined list or menu.**
2. **Keyword recognition-based chatbots: These are little complex where it tries to understand user types and accordingly respond.**
3. **Hybrid Chatbot: This is a combination of Keyword based and scripted reply chatbots.**
4. **Contextual chatbot: These requires data centric focus and hence, are the most complex ones and used Machine learning and Artificial Intelligence conversation and interactions with the user.**
5. **Voice enabled chatbots: This chatbot technology is future which uses spoken dialogue from user as an input that responses creative tasks. They make use of APi’s and Text to speech and the best example is Alexa for voice recognition chatbots.**

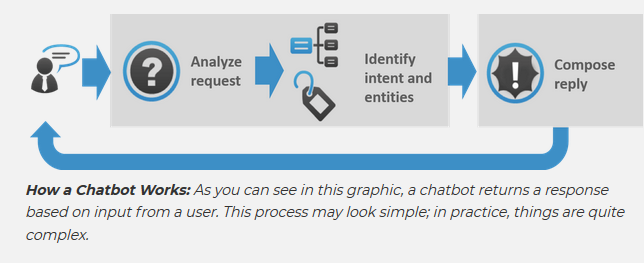
**3.3 Importance of Chatbots:**

A chatbot is often stated that it is most advanced and interactions expression within machines and humans. As per the technology point chatbot creates a session of question- answers between the machine and user by using natural language processing. Responding the user by using natural language is the actual example of that is applied in various sectors and fields.

Applications of chatbots are streamlining that interacts between services and people which enhances the customer experience. Simultaneously, they give new options to improve the process of customer engagement as well as reducing the typical cost of customer service by operational efficiency. To be convenient, chatbot should perform these tasks effectively. Human intervention is crucial in configuring, optimizing and training the chatbot system**.**

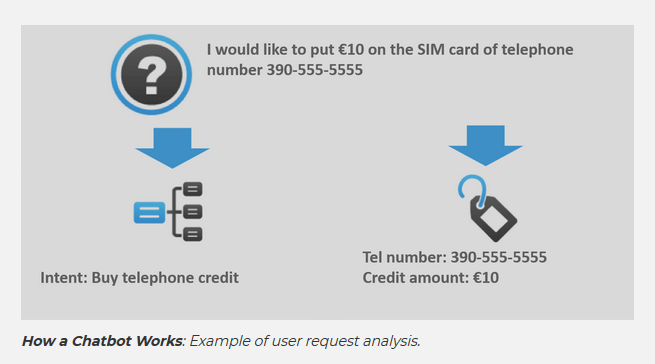
**3.4 Working of Chatbot:**

Working of a chatbot can be understood by the below mentioned image:



At the core of the chatbot, there are two different tasks:

* **User request analysis:** This is the first step of which a chatbot will perform which analyses the user request by extracting entities and intents. To understand and extract the entities and intents of the user which is the most relevant and first step.



* **Returning the response: After understanding the user’s intent, the chatbot is supposed to provide the response for user request. The replies from chatbot can be:**

1. **A pre-defined or a generic text.**
2. A text retrieved from a knowledge base that contains various solutions to user’s query.
3. Conceptualized information which is provided as per user’s data.
4. Action’s result which chatbot performs by interacting with backend application.
5. An unambiguous query which help the chatbot to understand user request.

**3.5 Purpose of Transport chatbot:**

Transport chatbot helps to book and cancel a cab by using phone number and postal code. It also included the fallback actions which is executed when the user tries to ask the questions that are not relevant. This means fallback action will be executed when the user is asking the questions that are out of the box.

**4.Scope of the project and tasks done by me and my team:**

We have implemented a transport chatbot which involves several things which are stated under:

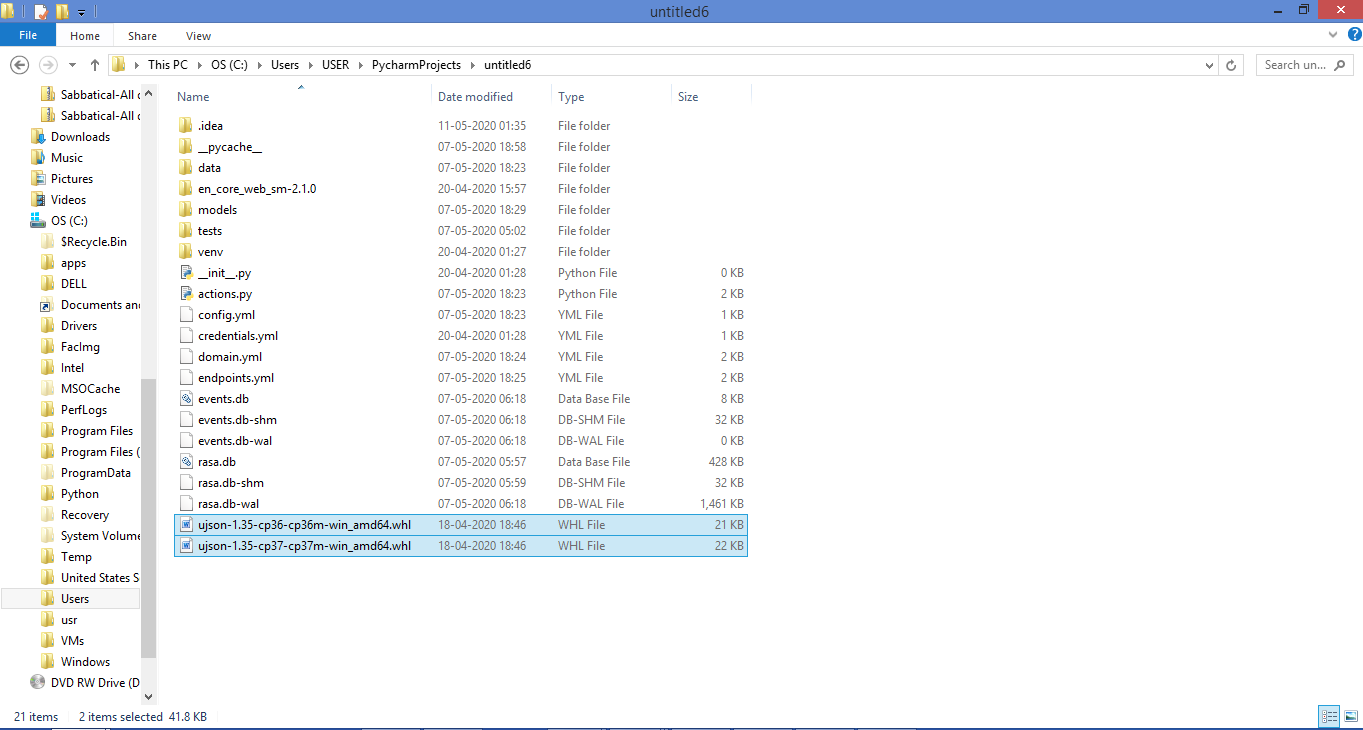
**4.1 Task done by me:**

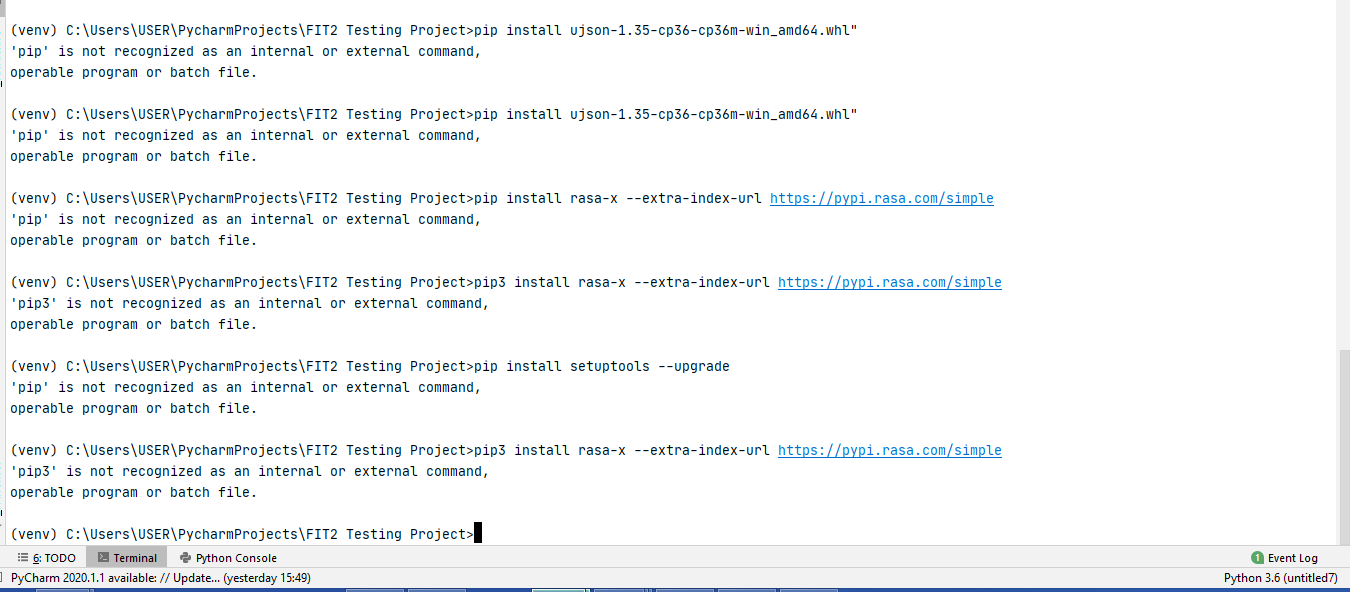
**Troubleshooting:**

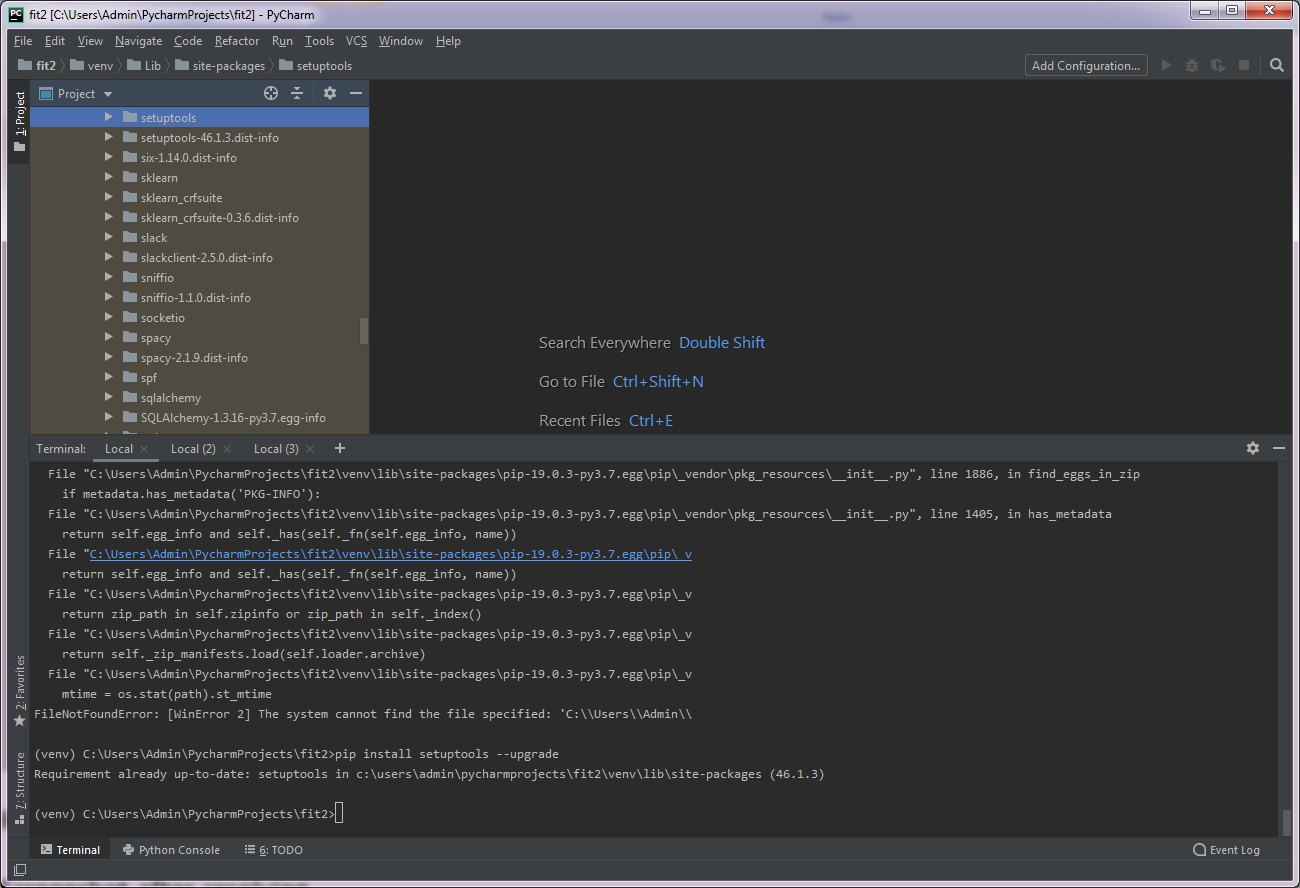
The chatbot that we implemented was a transport chatbot that will help user to book or cancel a cab. However, for building a chatbot there were some of errors which were massive and difficult to detect. Moreover, it took a long time to resolve those error and around 70% of these installation error resolved by me. Among such errors **DLL file not found,** .**whl package error and** **Spacy Folder** not found were some errors that I dealt and troubleshooted it in order to install rasa and they are explained below:

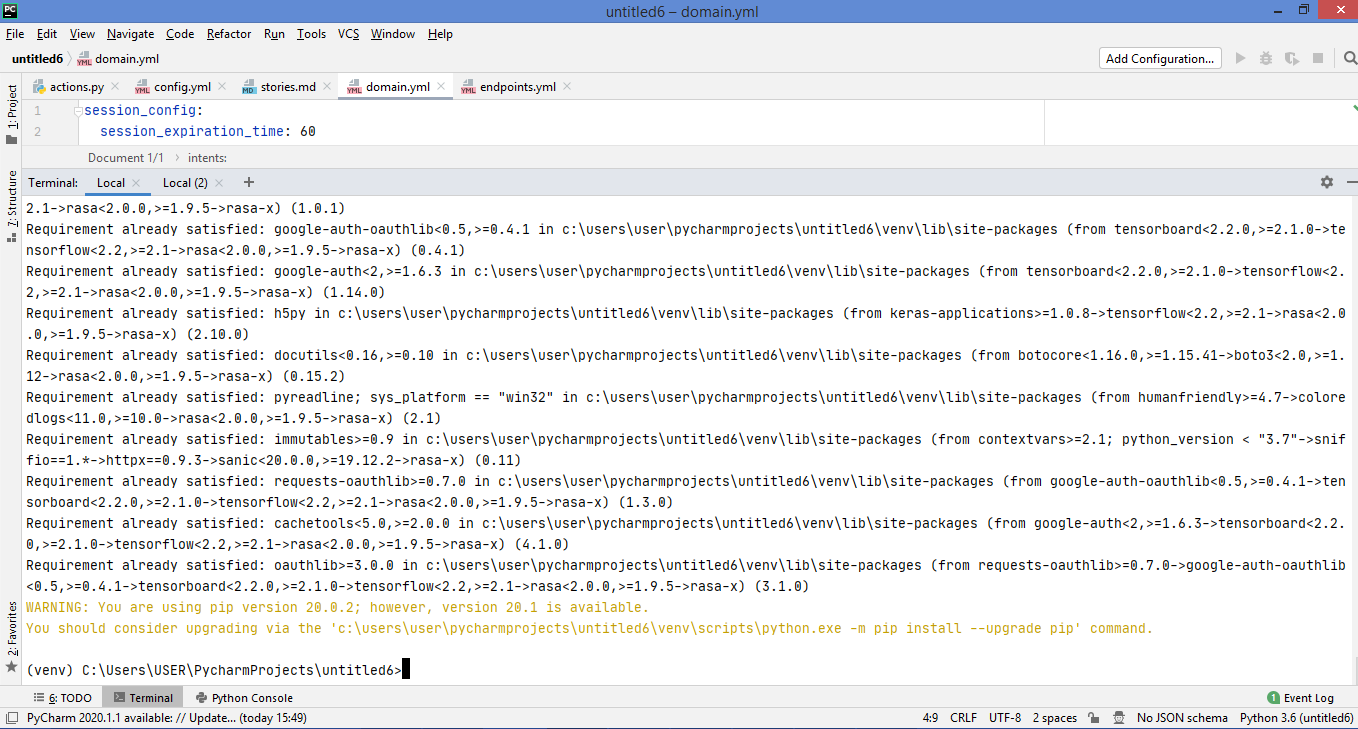
1. pip install rasa-x --extra-index-url <https://pypi.rasa.com/simple>

To install rasa using above command I was getting an error and tried many solutions but worked. At the end I tried downloading the downloading both the package .whl package (ujson-1.35-cp36-cp36m-win\_amd64 and ujson-1.35-cp37-cp37m-win\_amd64), but after that was getting file not found error which was resolved by “Pip install setuptools --upgrade” command







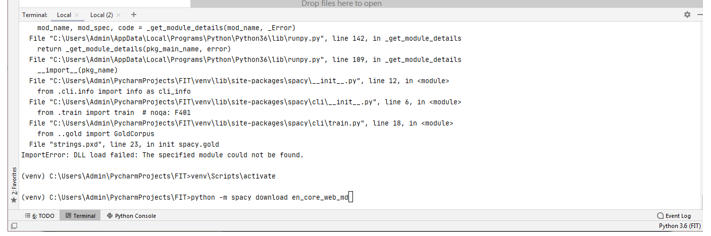


1. We need to download spacy but for this we were was getting an error like



For this error, used no cache command to clear history of ujson. Then, downloaded ujson wheel package as per the python version installed, if it is python 3.6.8 then we have to install the wheel version that relates to python 3.6.8 .

However, I was getting DLL file error after the above command, which was resolved by downloading the mcvcp140.dll and msvcp141.dll file in the site-packages folder or our project, and finally, I was able to download “en\_core\_web\_md”

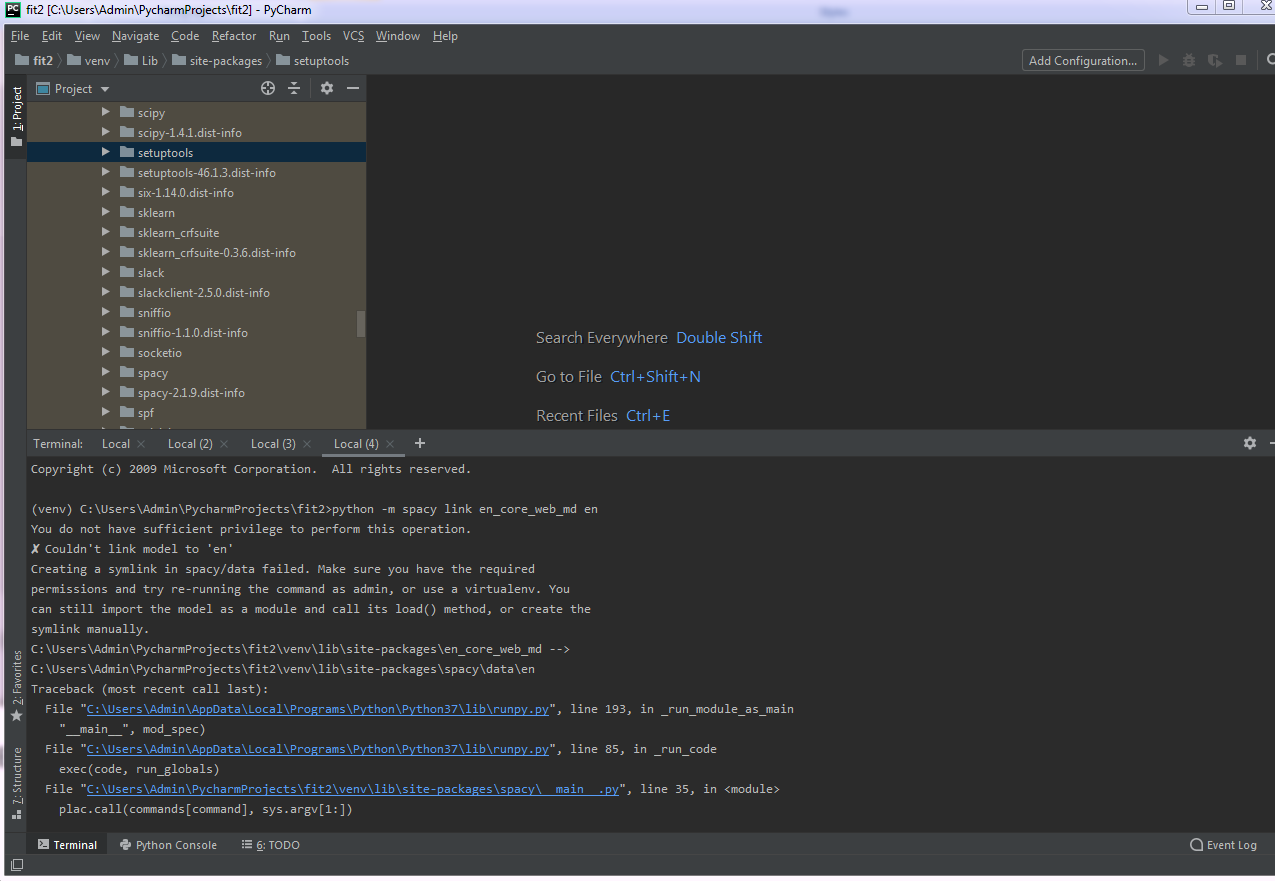


Then Finally, we were able to download spacy

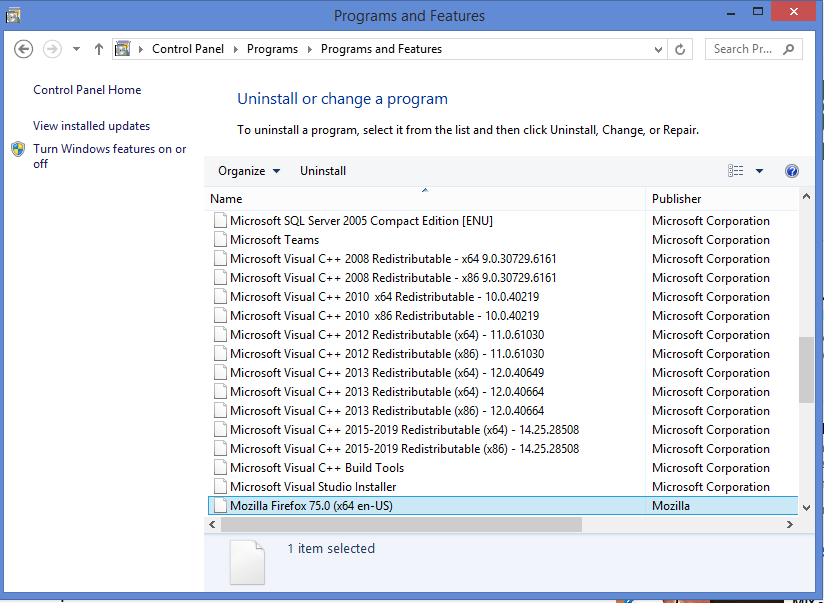
1. Next was to link a package by using command

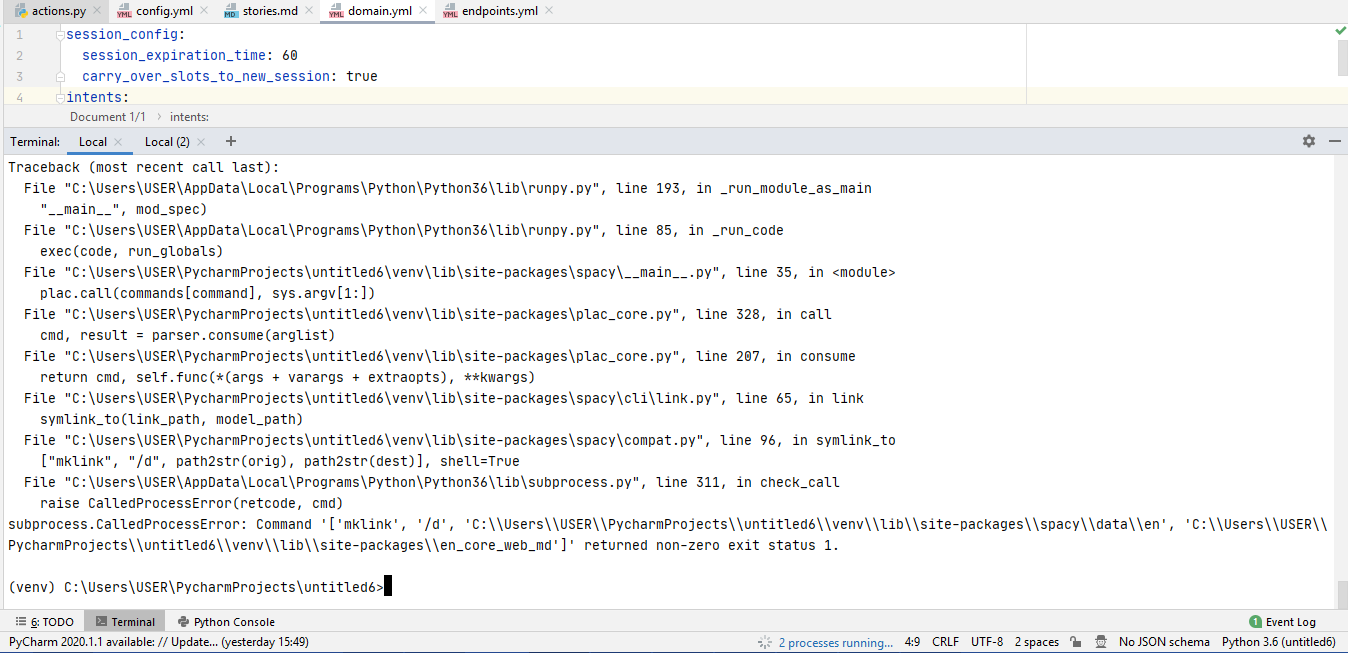
“python -m spacy link en\_core\_web\_md en”

But was getting below error:

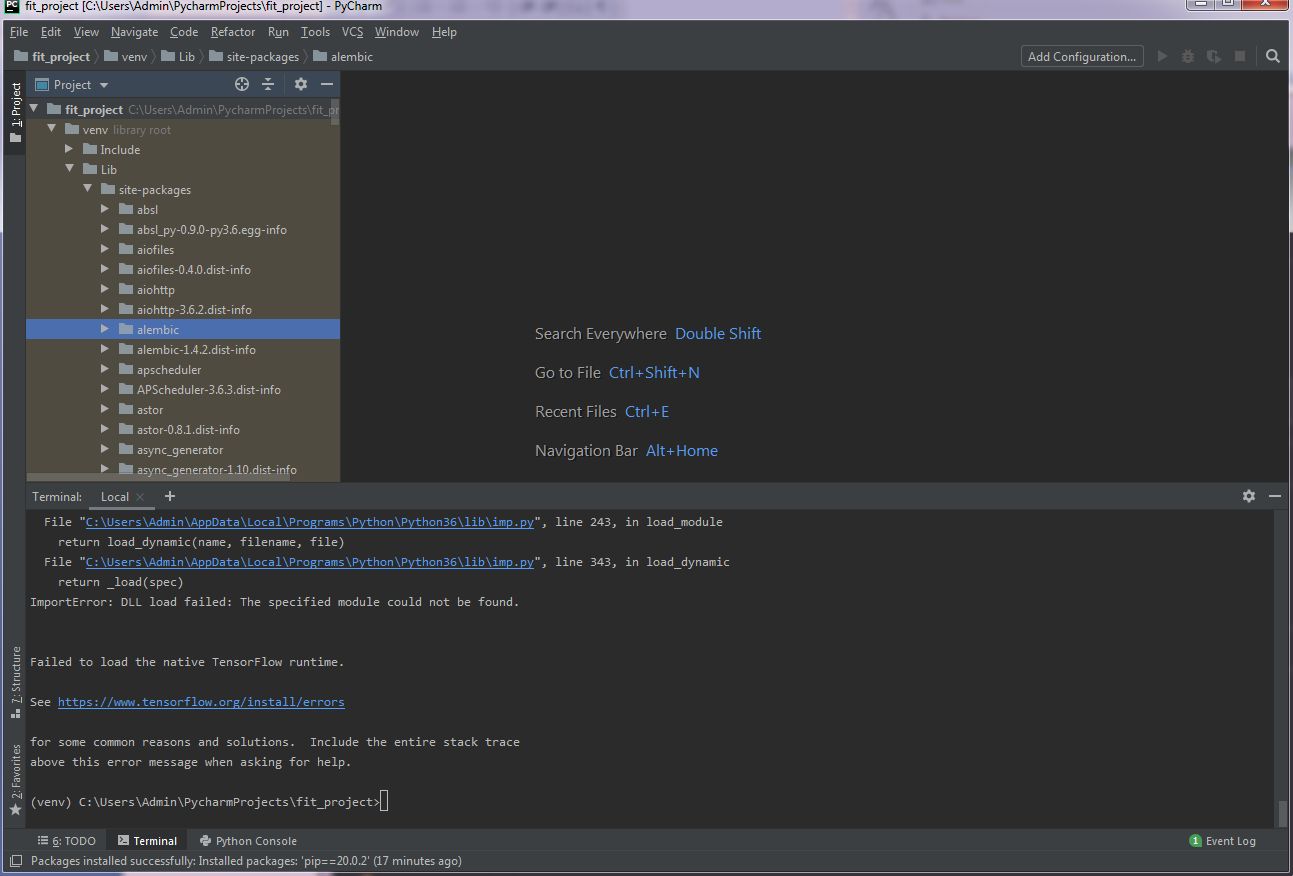


For this error, it took a lot of time and finally the solution was to create a shortcut en folder in site-packages folder. Then we created a shortcut en folder. However, we got one more error and the solution for it was to install the different versions of Microsoft Visual C++ application, then we were able to link en model

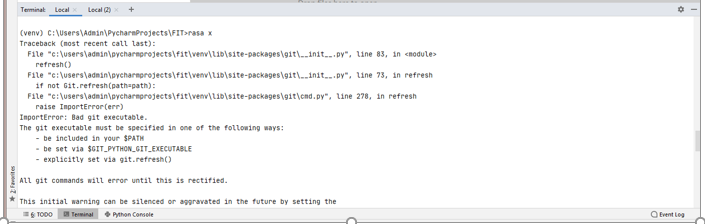


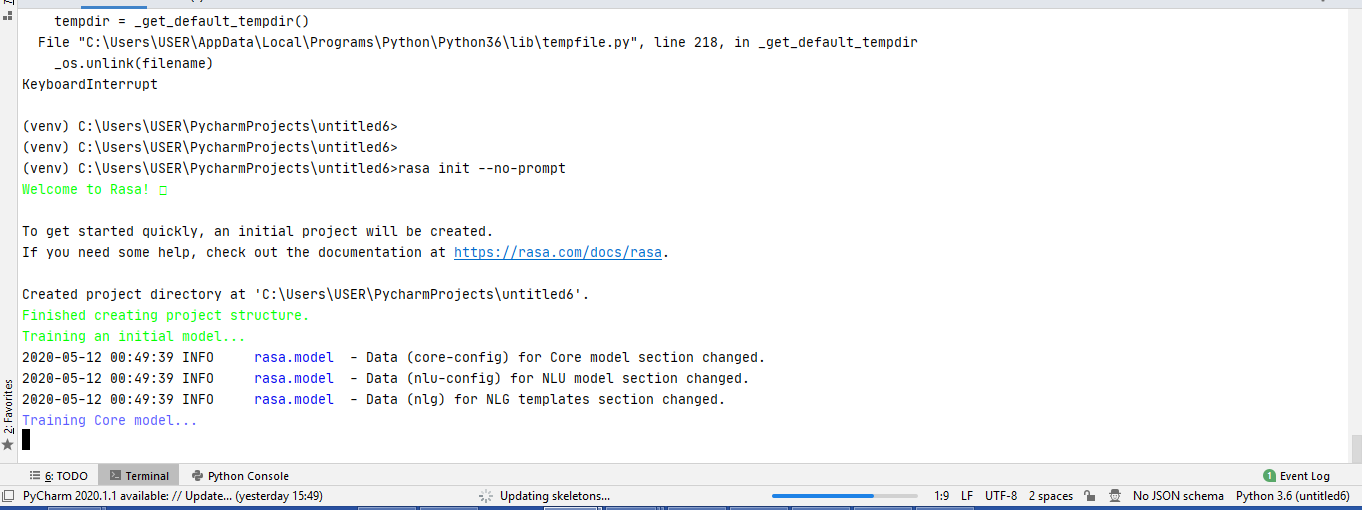


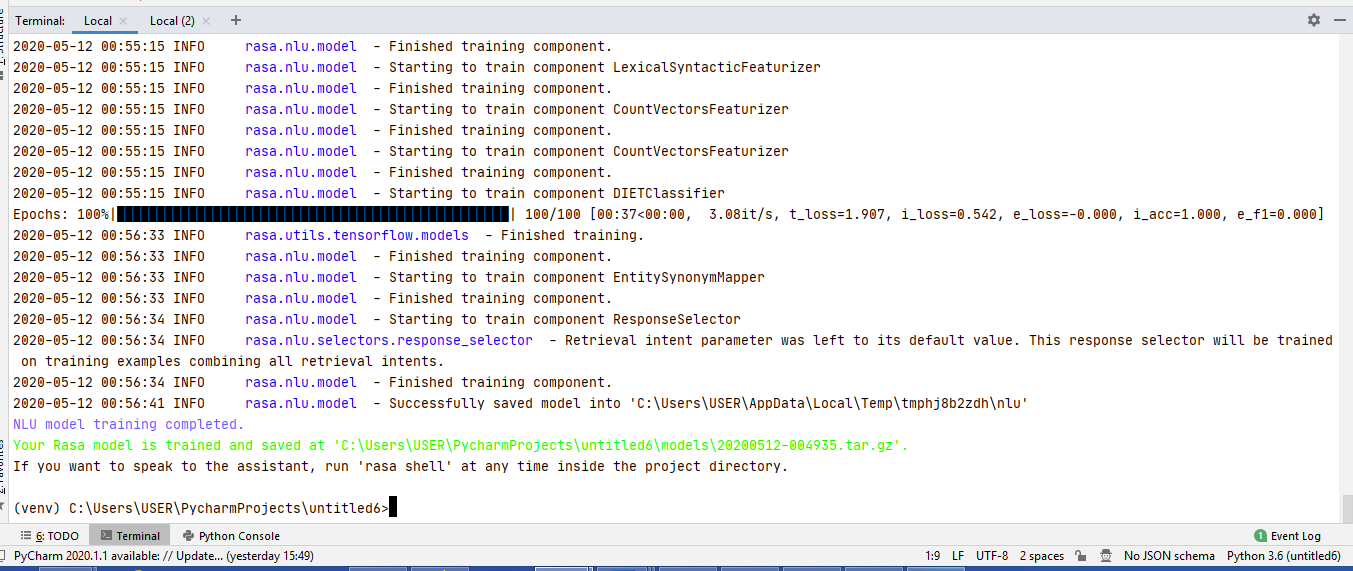
After that was getting error for native tensorflow while initiating rasa



But even after copying dll file was getting error after typing rasa x -and it was resolved by downloading Git and adding it to PATH environment variable. PATH - C:\Program Files\Git\bin\git.exe







Hence, all the above-mentioned installation errors troubleshooting was covered by me.

Also, I came up with the suggestions of ideas and scenarios like what scenarios will be included which were booking and cancelling cab with mobile number and postal code and including error handling scenario as well. Attaching the scenario sheet that I came up with



**4.2 Task done by team:**

**Tejaswini:** While thinking of the topic, Tejaswini came with the suggestion of Building a Chatbot by using Rasa and the necessary tools which should be used for creating a chatbot.

**Divya:** Almost 70% of the coding of file like Nlu, stories, domain, configuration, actions and Endpoint file was done by Divya

**5. Framework:**

An open source AI tool is Rasa Stack and because of this it is easier to customize. Moreover, there are many cases when used does not wishes to share some personal data and most of the tools provide Software-as-a-service which are available on cloud. Because, we cannot run it on internal environment and hence user need to share it to the third party. However, this is not the issue with RASA, where we can easily build, develop and deploy internally or on the server on which user can have the complete control.

**5.1 Components of Rasa**:

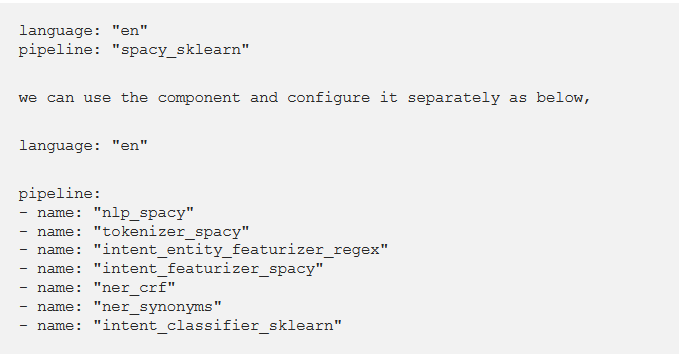
There are two major components of RASA

* Rasa NLU and
* Rasa Core

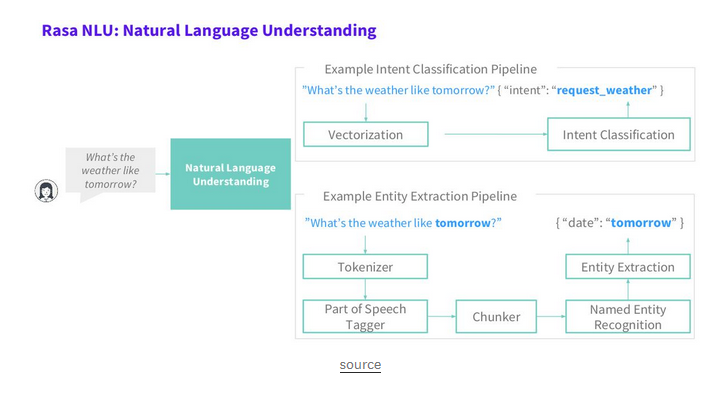
a) Rasa NLU: It translates user’s language into Natural processing language. It makes a differentiation between intent and entity and, extracts the entity from the input given by user that helps bot to understand user’s intent.

NLU is module that consists of loosed coupled module which has many natural language processing and libraries of machine learning that are in API. Some of pre-established pipelines are spacy\_skylearn, mitie, mitie\_sklearn, tensorflow which have defaults that are sensible and that works as expected in many cases. For an instance- most frequent used spacy\_sklearn uses following components to processes the text. Firstly, text is tokenized by using spaCy library, different parts of speech are annoted. Next, the featuriser of spaCy searches for Glove vector for all the tokens and pools for creating a whole sentence representation. After that Scikit\_learn train estimators for dataset, however, default trained multi class support vector is 5-fold cross validation. Then ner-crf trains a field that is random and identifies entities of training data, by making use of parts of speech tags and tokens as base-features. However, all components make use of same API’s, it is simple to exchange the custom vectors of GloVe, embeddings of domain specific words or using other library of machine learning for training the classifiers.

Example: spaCy is used as a template by spacy\_sklearn and is stated as under:



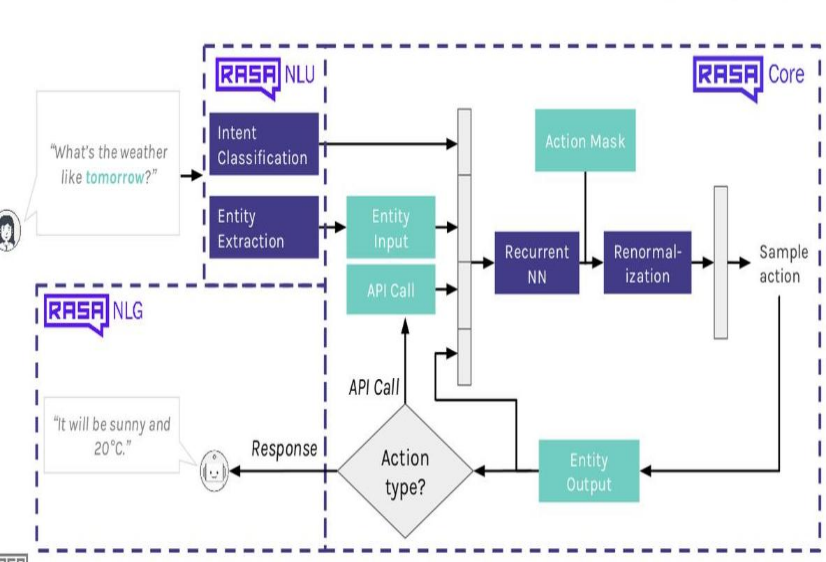
Entity and intent identification is done as stated below in the diagram:



b) Rasa Core: It is a chatbot framework that has dialogue which is based on machine learning that takes input from NLU and makes the prediction of new best action by making use of a model which is probabilistic like a neural LSTM network.

An action can be a simple utterance, i.e. message sending to the user, either it can be an arbitrary function which can be executed. While executing an action, it first passes to a tracker instance, and hence uses relevant information which is collected from the dialogue history: previous utterances, previous actions results and slots. Actions are not directly mutated from the tracker, but while executing actions can retrieve events list. The events are consumed by the tracker which updates its state. There are various types of events, like AllSlotsReset, Restarted and SlotSet etc.

Below diagram states the working of RSA NLU and RASA Core together:



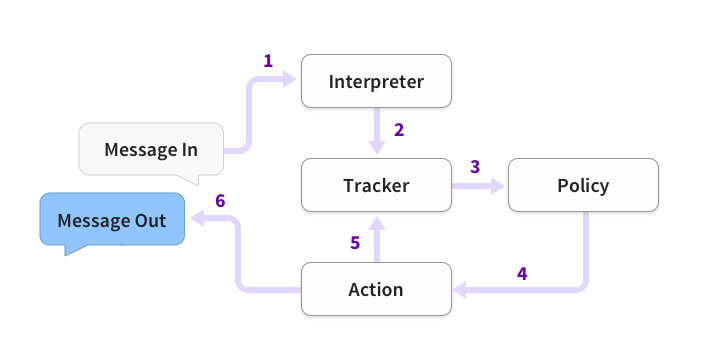
Both core and NLU are not dependent on each other and use can use either of both, however, we have made use of both.

**5.2 Architecture of RASA:**

Rasa architecture involves six steps that start with receiving the messages and responding to user’s query:

* The user sends the message and once the message is received it is passed to an Interpreter, which converts into dictionary which includes actual text, user’s intent, and the entities in user’s intent.
* Then tracker keep a record of the conversations made and receives the notification that new message is there.
* The policy then gives the present state of tracker.
* The policy makes the decision which action to be next.
* The new chosen action is tracked in the tracker.

Finally, in the end, a user receives a response



**5.3 Why Rasa over other platform:**

There were various platforms available to build a chatbot, however, we used RASA not only because it is openly available but there are many other reasons. Rasa is NLU based dialogue management and allows users to interact in a fruitful way. By making use Rasa, it is easier to build controversial AI that can be improved over time. The best feature of Rasa is it uses interactive learning methods that is easier to understand and saves money, resources and time. Moreover, rasa claim that the standard infrastructure for AI chatbot is provided by which makes easier for companies and developers to create a chatbot of their own requirements. Moreover, there are many other reasons to choose rasa and they are easy to integrate and customize, no state machines, integrate into existing systems, run it on your favorites, supports several intents, interactive learning, connects with other messaging app, multiple deployment environment available, role based access control and Analytics an d reporting. So, these are the reasons why we choose Rasa over any other framework.

**5. 4 Intent Classification Architecture:**

Intent is query from the user and for that a chatbot uses pattern-based approach. However, the challenge for using pattern-based approach is the patterns are supposed to be coded manually which is not that easy. This means that sometimes we ned to hardcode all the answers the chatbot can give which is terrible and difficult and, causes overlapping confusion between the patterns. This can reduce the chatbot accuracy. Also, it is difficult because the same questions can be asked in multiple ways. Hence, it is hard to define the pattern by natural language understanding, but computers can do that easily. In simpler terms a small domain can use a pattern architecture but if a chatbot deals with different domains and many services, state of art neural network like reinforcement learning agents and Long Short-term Memory are the best option.

For intent classification, machine learning can be applied that classifies and rectify the patterns of natural language. For this we just need to give some examples from which it will choose the pattern for the user request and make the intent classification fairly and accurately in minimum amount of time.

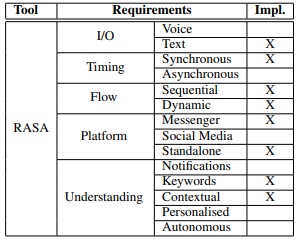
These types of machine learning algorithm can be developed by using library of machine learning such as Tensorflow, PyTorch, Sci-kit learn. Moreover, a chatbot uses machine leaning algorithm to respond to the user which ultimately uses the library of tensorflow.

**5.5 RASA X:**

Rasa X is a GUI tool which is browser based and allows to train machine learning model by using interactive mode of GUI, however, it is optional. Rasa sends information statistics from the web, but it never sends training data to outside systems, however, it only sends the number of times Rasa X is used. It helps you to build, improve and deploy AI assistants which are facilitated by RASA framework. It has a Rest API and a user interface. Rasa x is the latest release from Rasa.

**5.6 Features provided by RASA:**

RASA has a library which is open source that covers understanding of natural language and the management of dialogues. Below mentioned table shows the rasa requirements which can be implemented on RASA. It states that RASA can handle text inputs directly but not a spoken language.

****

**Table 1: Capabilities provided by RASA**

**5.7 Description of Chatbot pipeline:**

On the basis on RASA-NLU, a transport chatbot is build and the interactions between user and bot is through Pycharm debugger and Google Chrome browser. Once the user tends to book or cancel the cab, message is sent to the bot and then NLU précises the entity from the message. Finally, chatbot understands user’s intent and then responds to the queries. In this, we have chosen spaCy model for “en\_core\_web\_md” for language model. The query from the user is group of messages for which user expects the response which is annoted with the entities and intent which NLU extracts. In our case it includes greet, I\_need\_to\_book\_a\_cab, How\_many\_People, postalcode, phone\_number, Thank you, I\_need\_to\_cancel\_a\_cab, cancelcab, cancel\_phone\_number, bot\_challange. In this, the below mentioned intents are included in the ex of dataset “greet”, “I\_need\_to\_book\_a\_cab”, “How\_many\_People”,”postalcode”, “phone\_number”, “Thankyou”,” I\_need\_to\_cancel\_a\_cab”, “cancelcab”, “cancel\_phone\_number”,”bot\_challange”.

Then we have trained the model by using “Rasa Train” command.

**6. Coding and Implementation:**

**6.1 Coding:**

There are various types of files that needs to be edited as per the specific requirement and they are Actions.py, Config.yml, domain.yml, Endpoint.yml, nlu.md and Stories.md. All the above six file

are available for different purposes. They are as mentioned below:

* **NLU.md**: This file has all the intents and entities.
* **Domain.yml**: This file includes actions, entities, intents and templates.
* **Config.yml:** This file includes config for configuration of nlu and core models.
* **Actions.py**: This file includes actions for coding actions which has actions and includes policies and pipline that is defined as supervised\_embeddings
* **Endpoint.yml**: Endpoint file have details related application link and all.
* **Stories.md: This file** defines stories and the extension of file ‘md' stands for markdown. Markdown is the easiest and most readable Rasa NLU format.

All the above files are attached under:

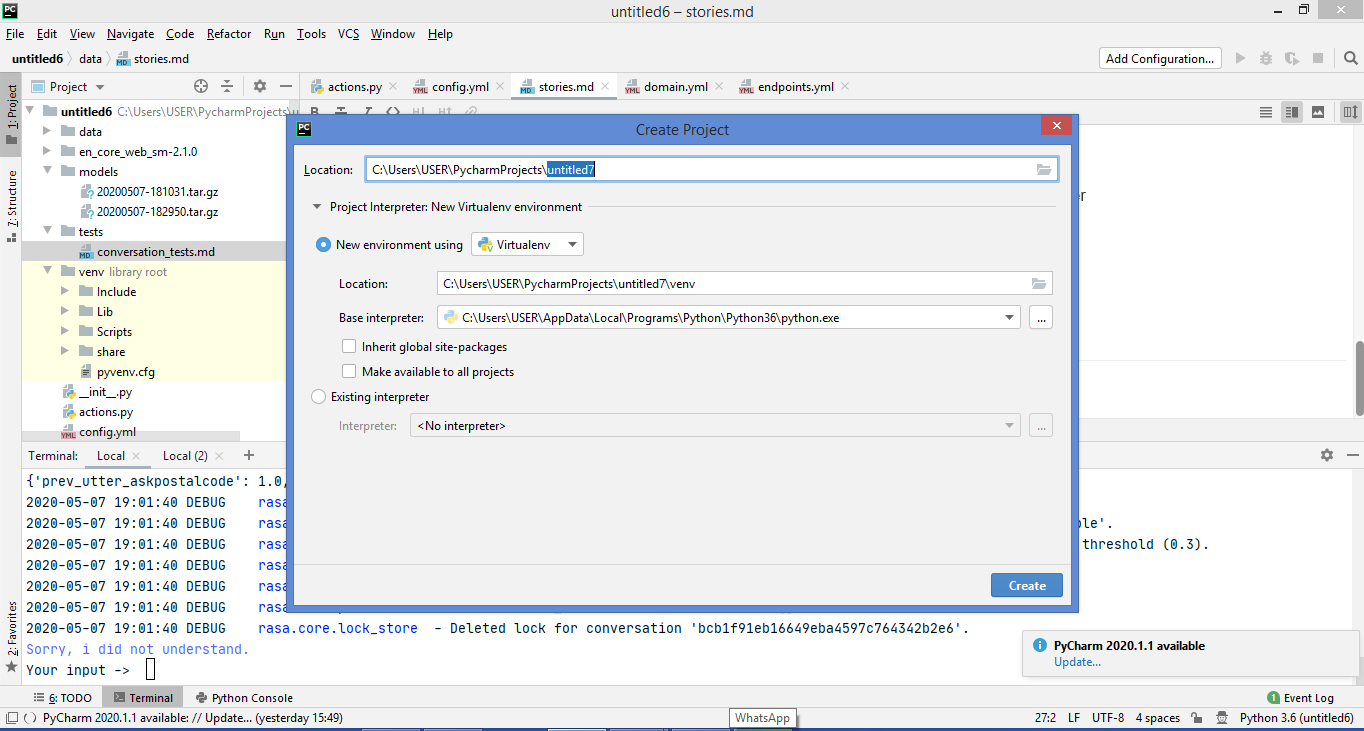
****

**6.2 Implementation:**

In order to install Rasa, we have installed following mentioned application before we ran installation commands for Rasa installation:

* Pycharm
* Python
* Microsoft Visual Studio
* Microsoft Visual C++ Build tools
* Microsoft Visual Studio C++ with different versions.

Once the above software is available, then we executed the below mentioned commands to create a new project and there were some errors while installations.

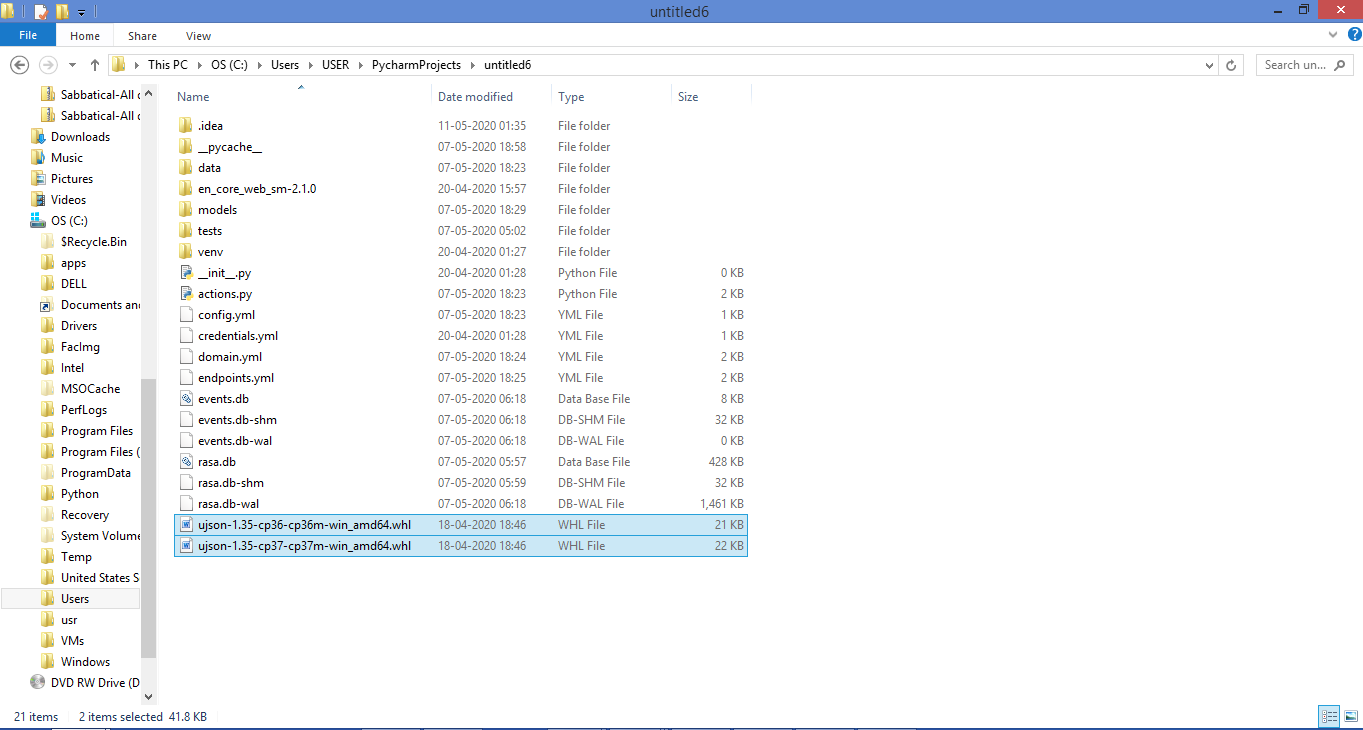


1. Venv\Scripts\activate:

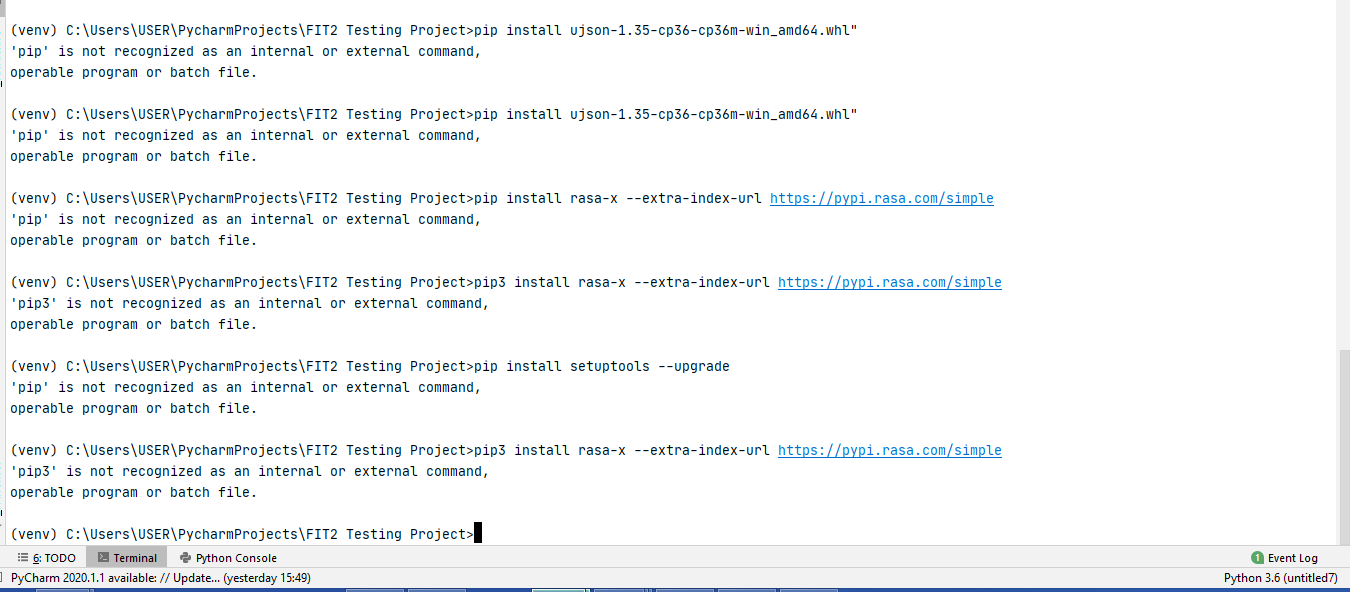
This command is used to activate the required scripts that are necessary for installing RASA

Next command after that is pip install rasa-x --extra-index-url <https://pypi.rasa.com/simple>. However, this command is not working for me. So, to resolve this issue I have used the Ujson package directly by using

1. pip install ujson-1.35-cp36-cp36m-win\_amd64.whl"

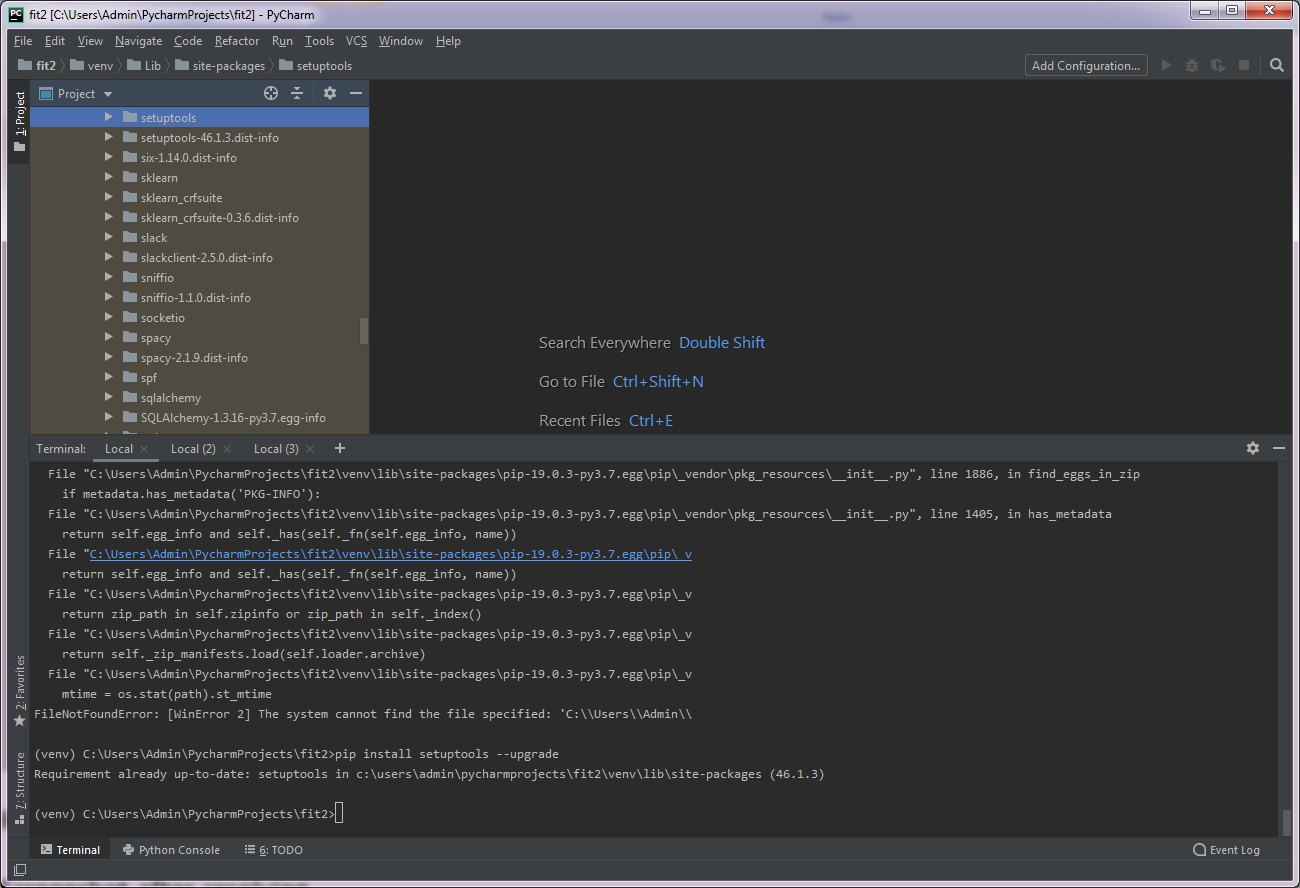


After installing both the .whl package (ujson-1.35-cp36-cp36m-win\_amd64 and ujson-1.35-cp37-cp37m-win\_amd64), then I was getting File not found



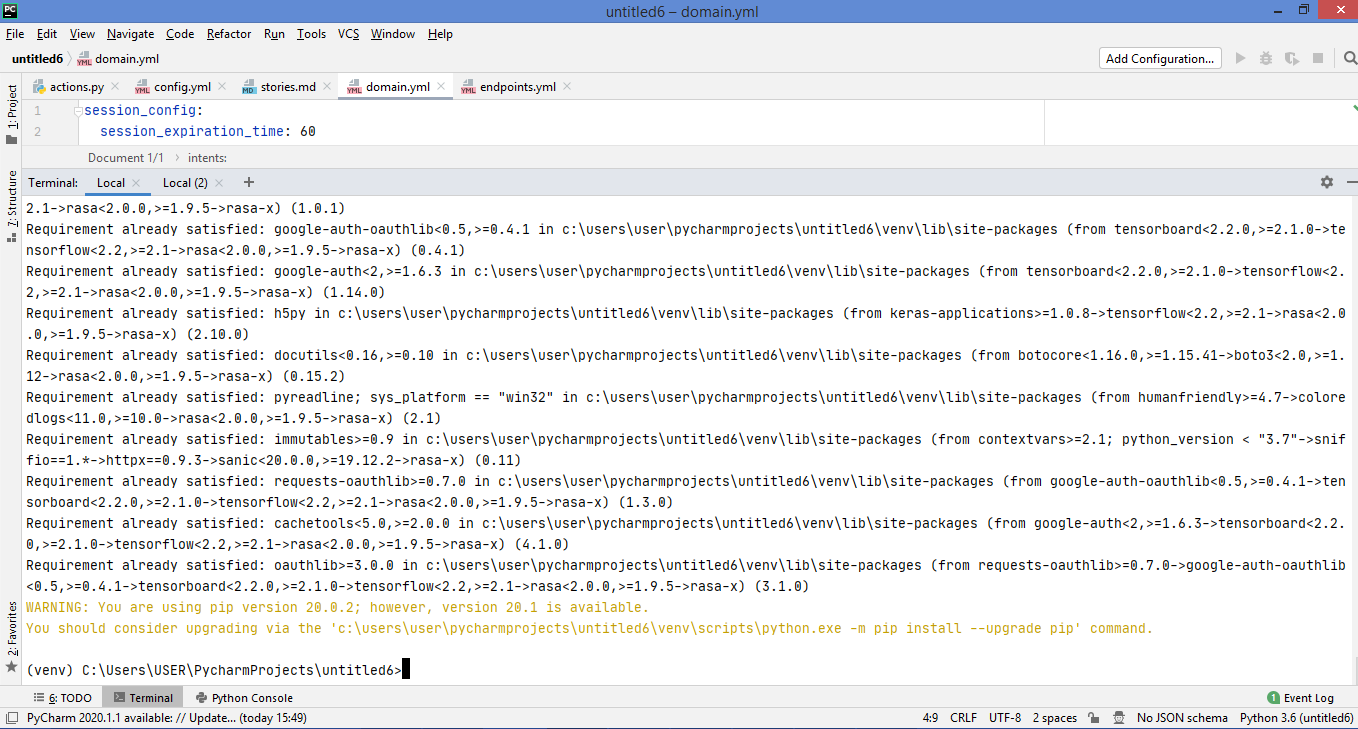
Then I resolved this issue resolved with

1. Pip install setuptools --upgrade



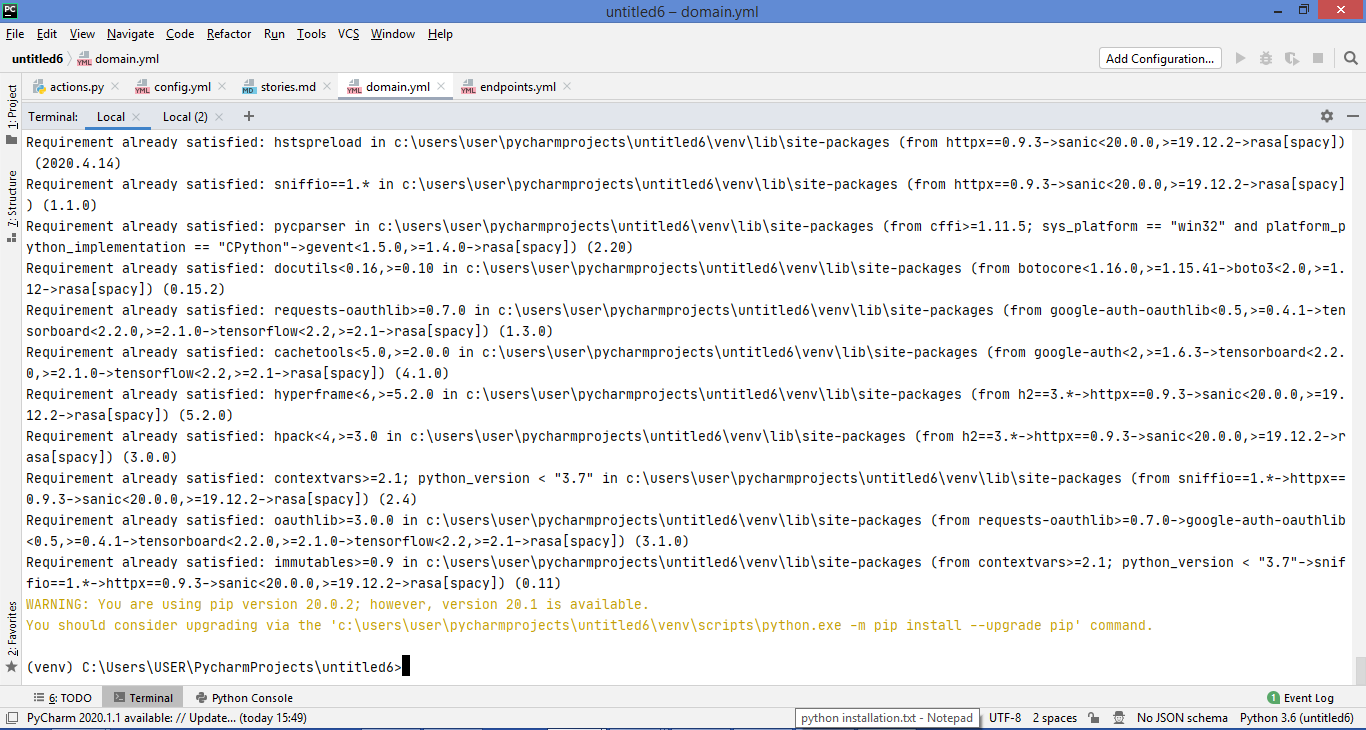
At the end, I was successfully able to execute the command.

1. pip install rasa-x --extra-index-url https://pypi.rasa.com/simple : This command installs all the necessary dependency and packages for Rasa.



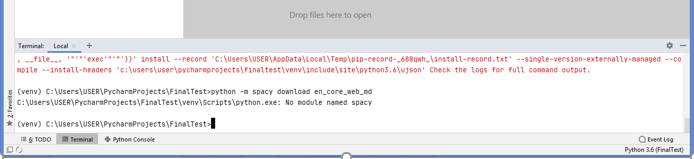
After successful installation of Rasa-x, Spacy needs to be downloaded by the below mentioned command:

1. pip install rasa[spacy]



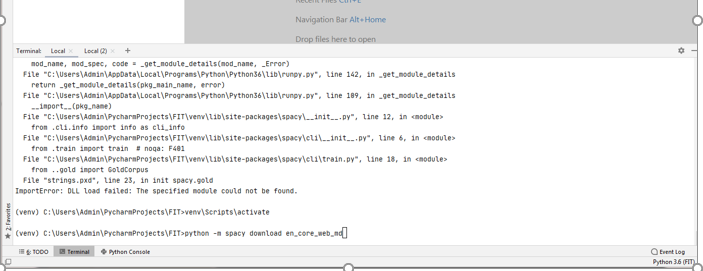
After installing Spacy, we need to download “en\_core\_web\_md” by using the below mentioned command:

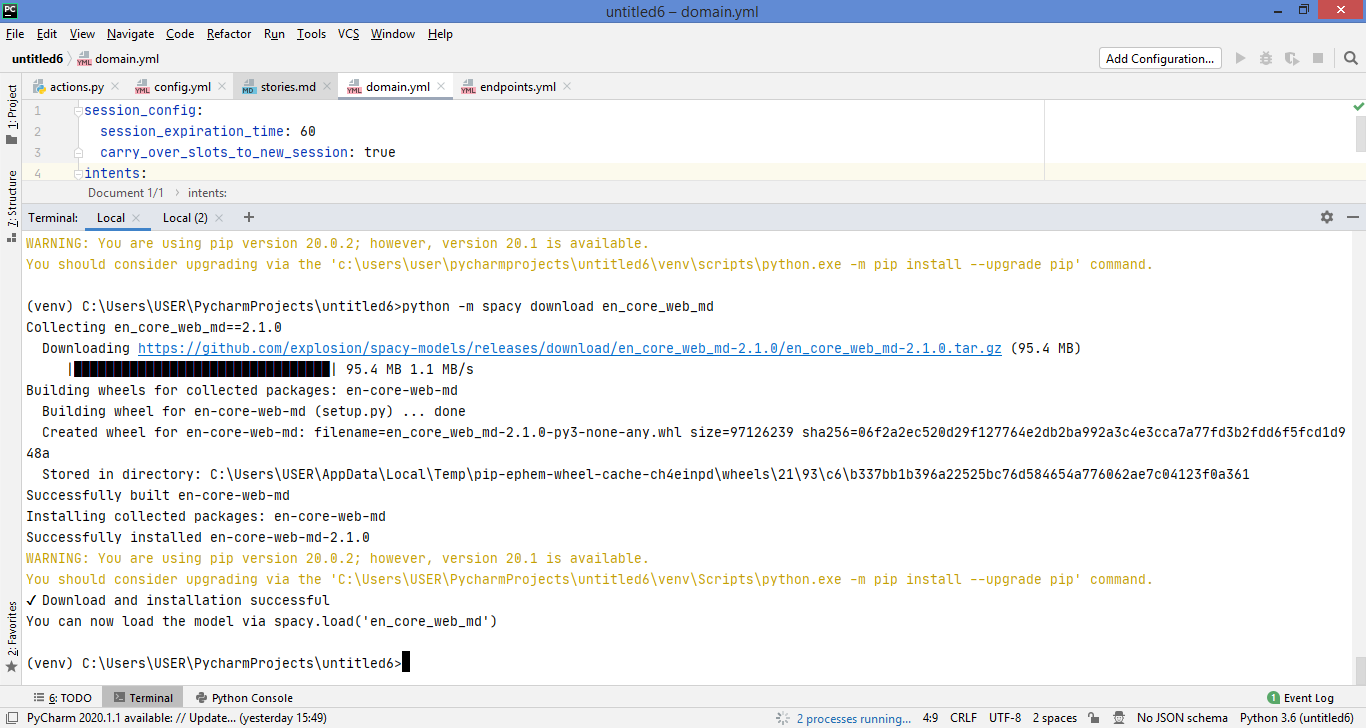
1. python -m spacy download en\_core\_web\_md



For this error, used no cache command to clear history of ujson. Then, downloaded ujson wheel package as per the python version installed, if it is python 3.6.8 then we have to install the wheel version that relates to python 3.6.8 .

However, I was getting DLL file error after the above command, which was resolved by downloading the mcvcp140.dll and msvcp141.dll file in the site-packages folder or our project, and finally, I was able to download “en\_core\_web\_md”

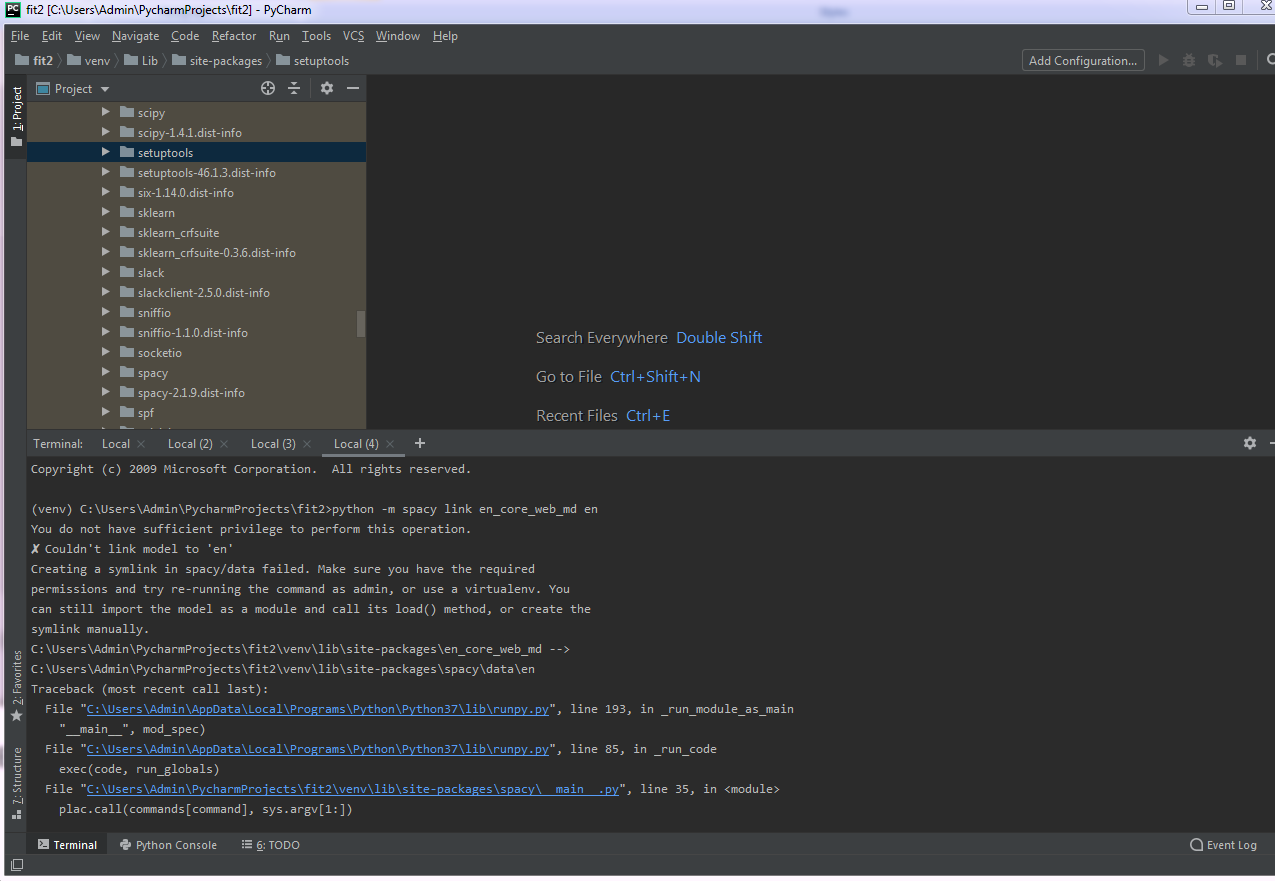




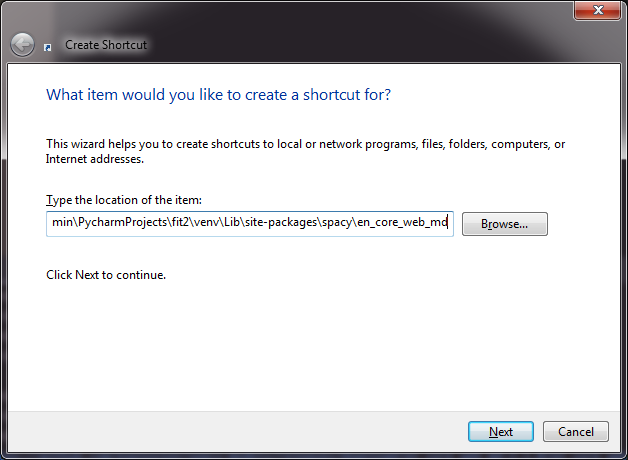
Next step is to link en model which is achieved by below mentioned commands:

1. python -m spacy link en\_core\_web\_md en.

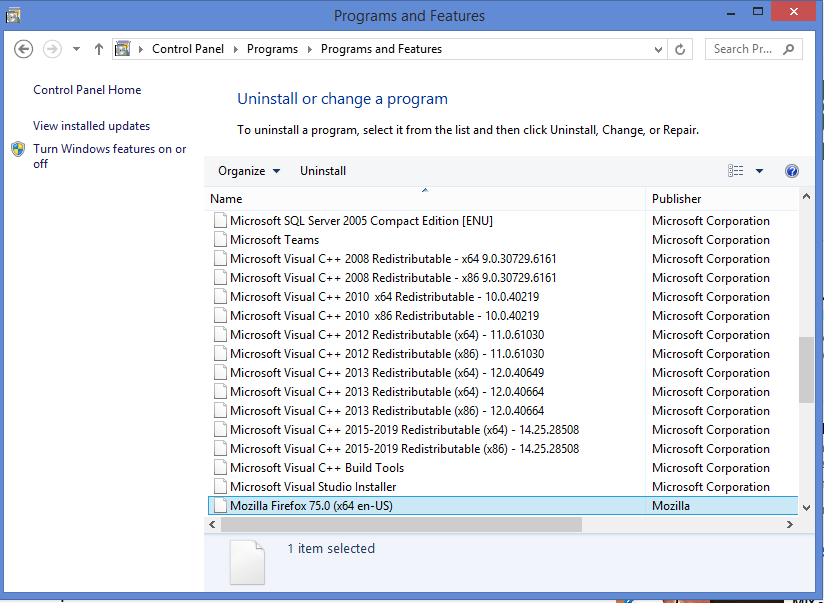
But was getting error, while downloading en

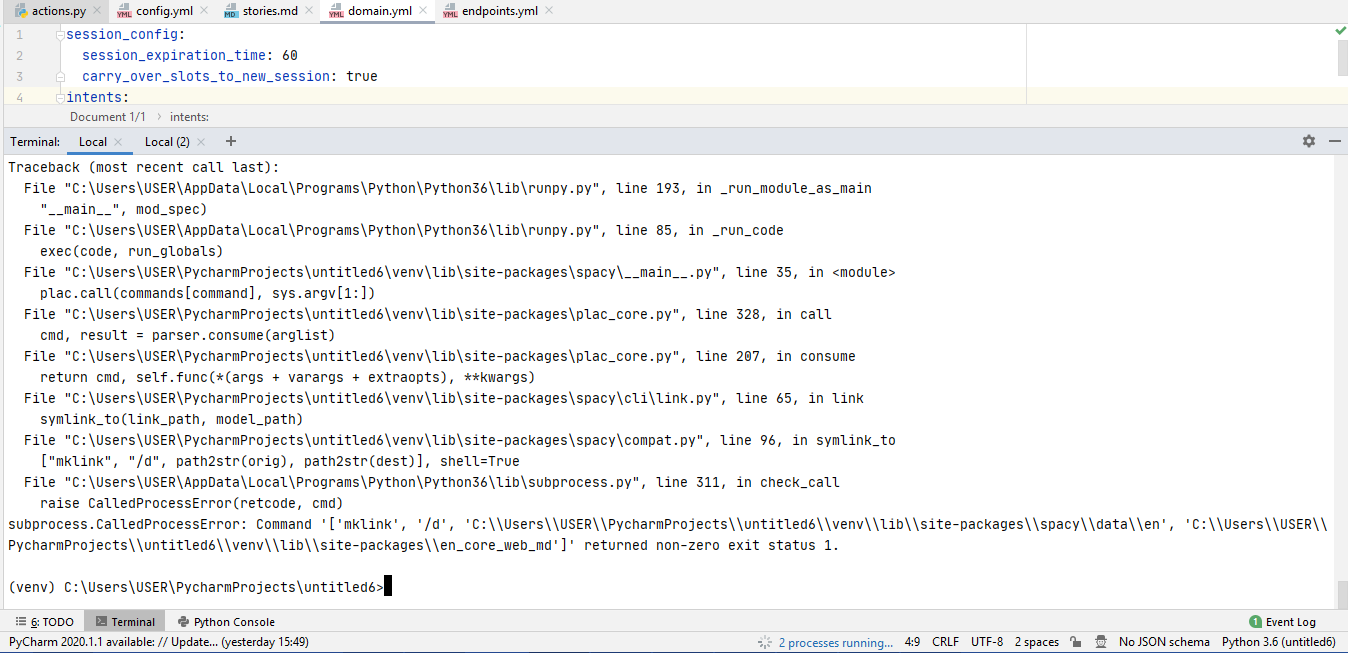


This was resolved by creating a shortcut of en folder in site-package:



After creating shortcut for en, still we were getting error and the solution for it was to install the different versions of Microsoft Visual C++ application, then we were able to link en model:

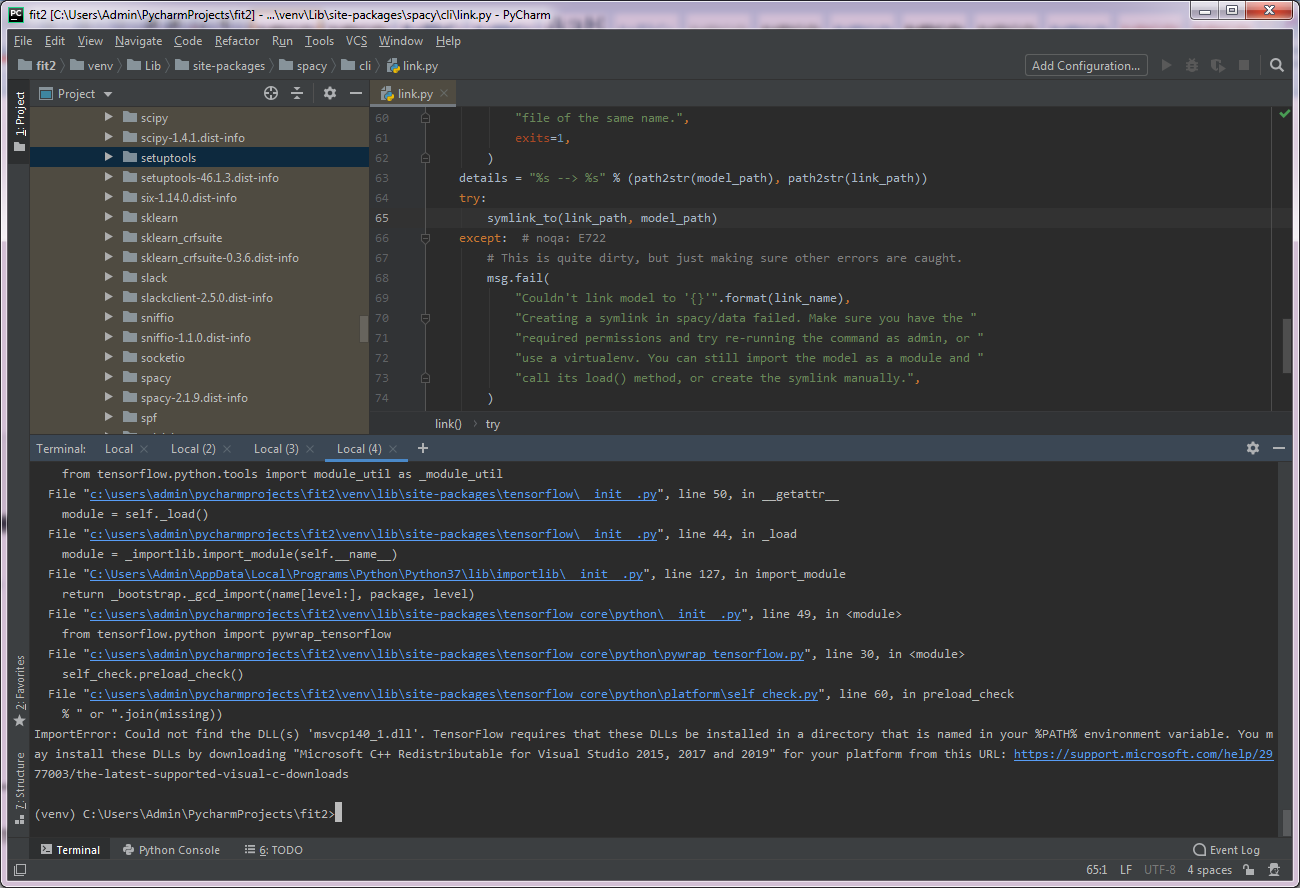




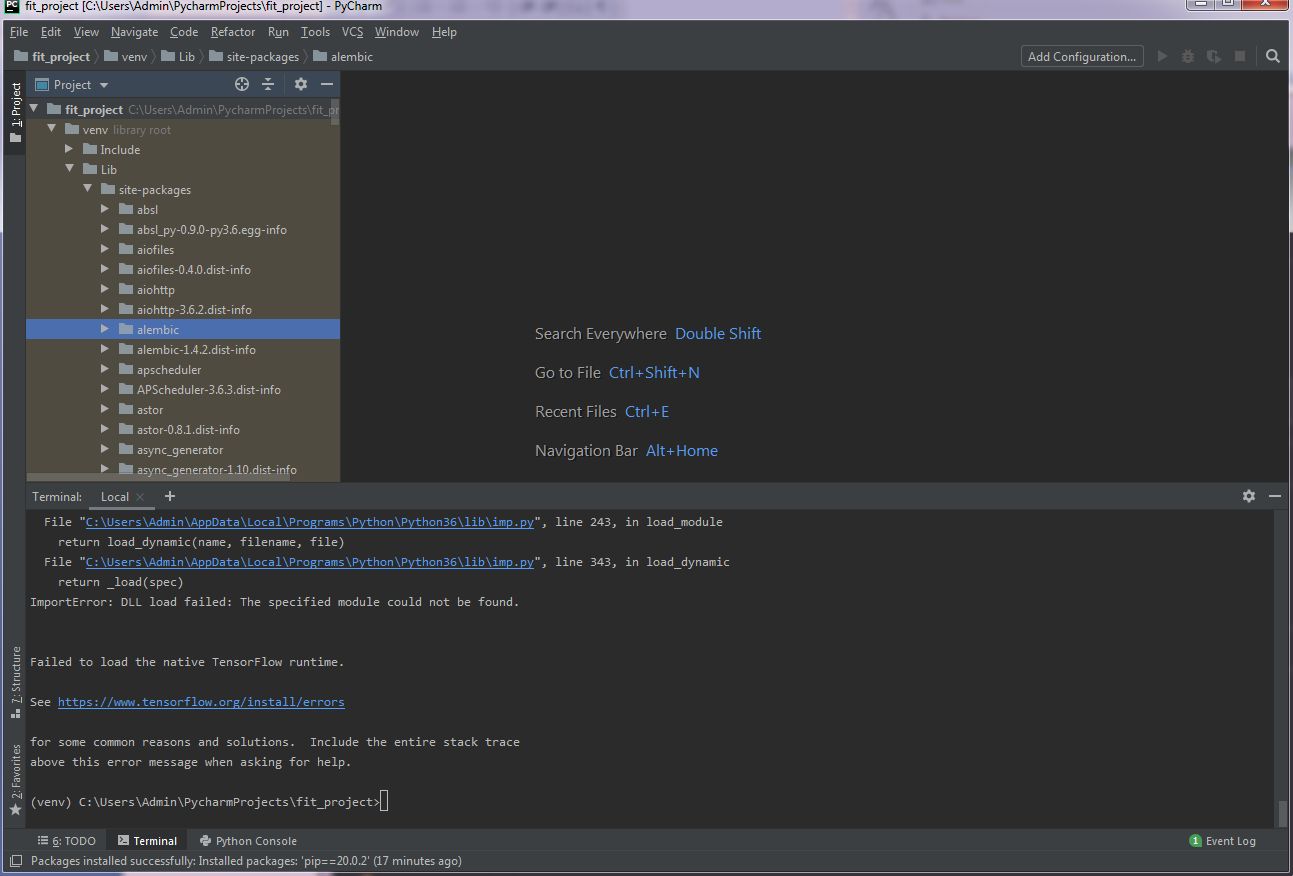
Finally, next step is installing Rasa x:

1. rasa init --no-prompt

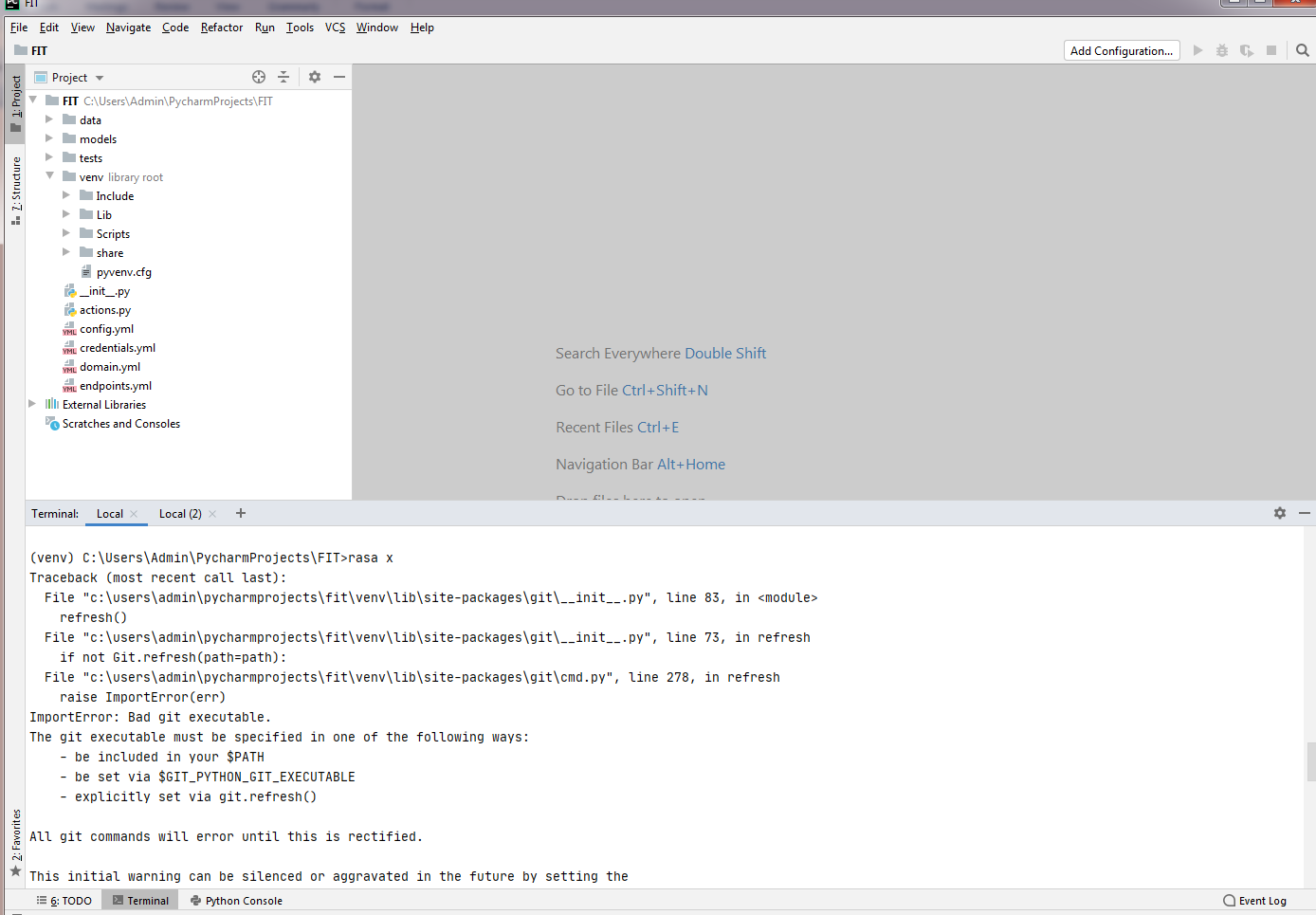
But was getting error while initiating rasa, resolved by copying msvcp140\_1.dll to project->python36 location

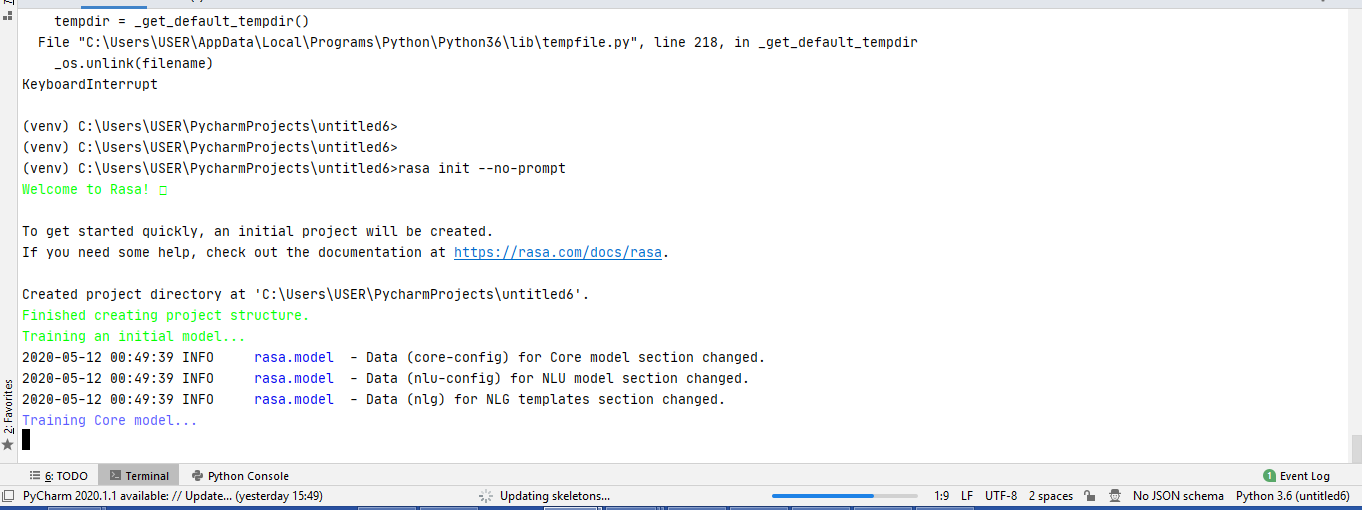


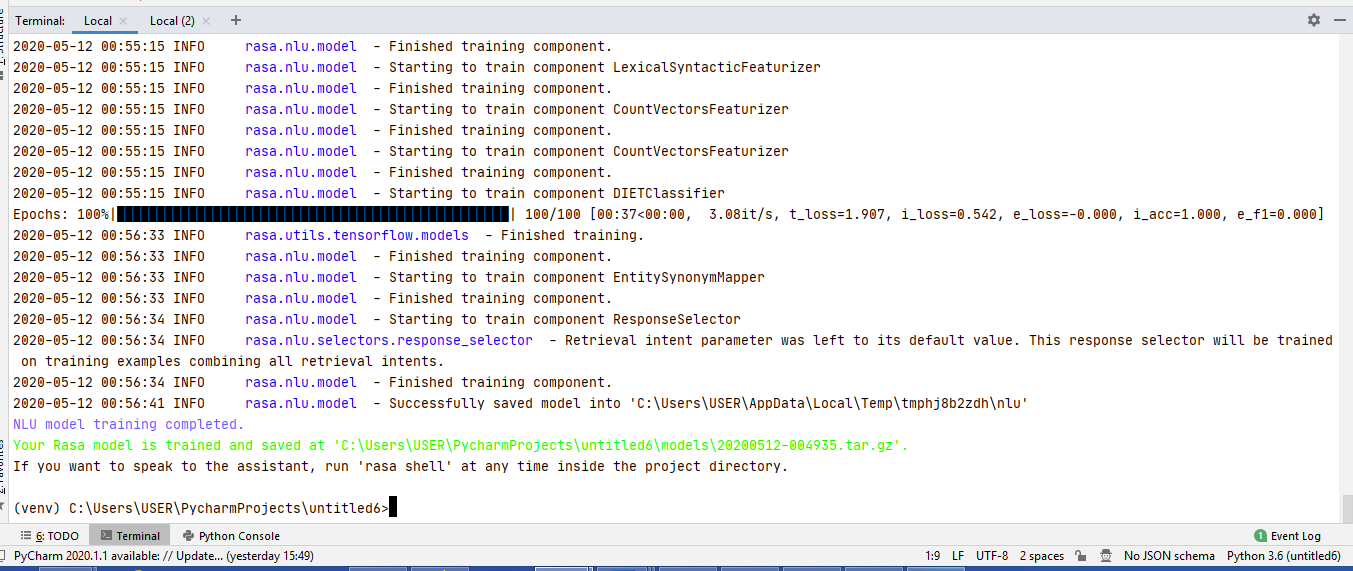
After that was getting error for native tensorflow while initiating rasa



But even after copying dll file was getting error after typing rasa x - resolved by downloading Git and adding it to PATH environment variable. PATH - C:\Program Files\Git\bin\git.exe

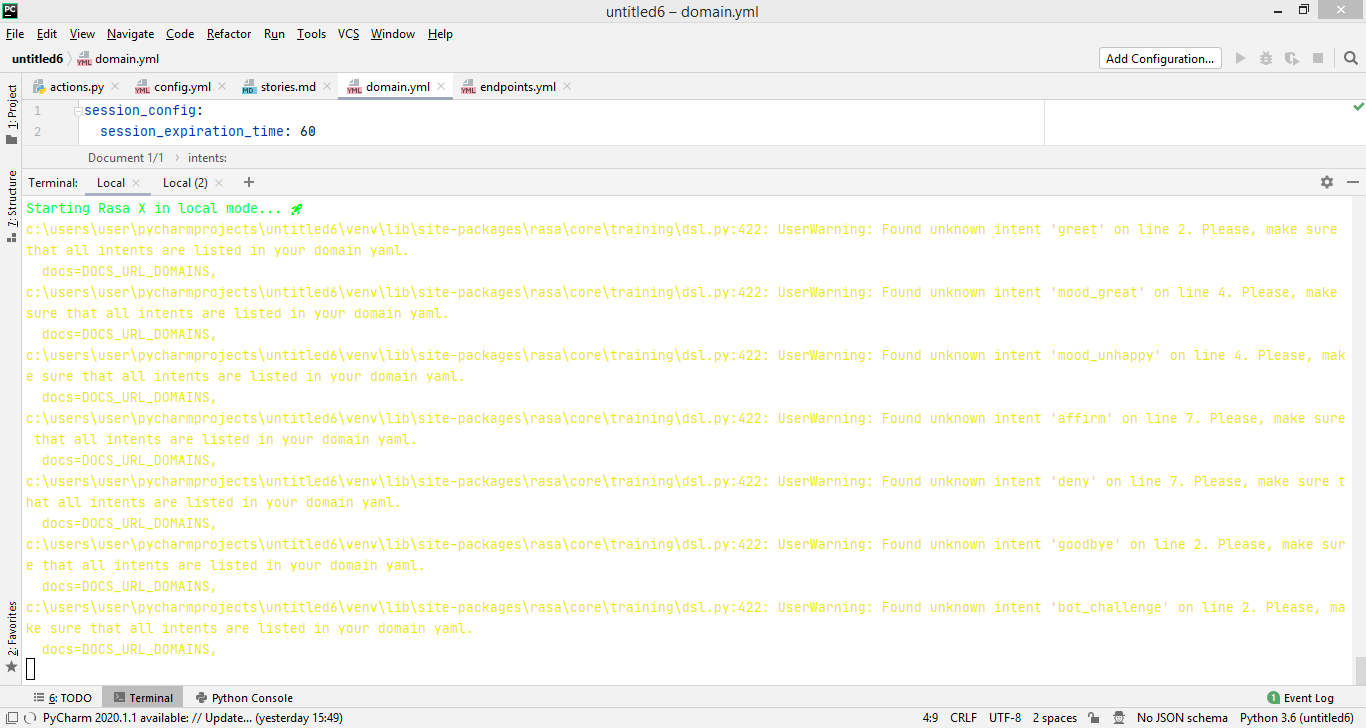


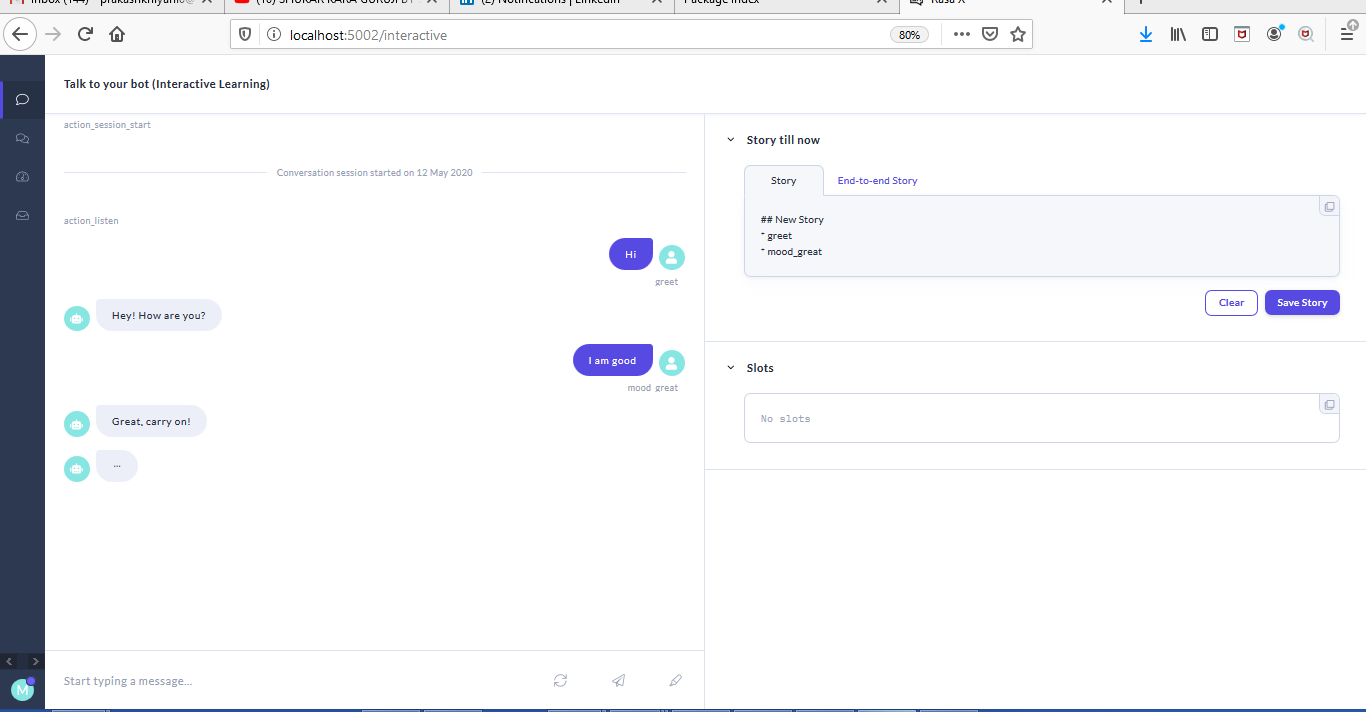




After installing the Rasa x, now we need to open the chatbot by using below mentioned command:

1. rasa x





**7. Execution:**

**7.1 Execution part:**

After executing Rasa x command, the window for a chatbot launches which helps the user to communicate to chatbot. After the launch of chatbot we have modified the files like nlu.md, Config.yml, domain.yml, Endpoint.yml, Stories.md and Actions.py as per our code. After saving those files we need to train our model by using “Rasa Train” command which creates a .gz file in model file in our project. After that we need to run the chatbot which can either be in the debugger window or in the browser.

To run the chatbot in debugger we need to execute the below mentioned commands:

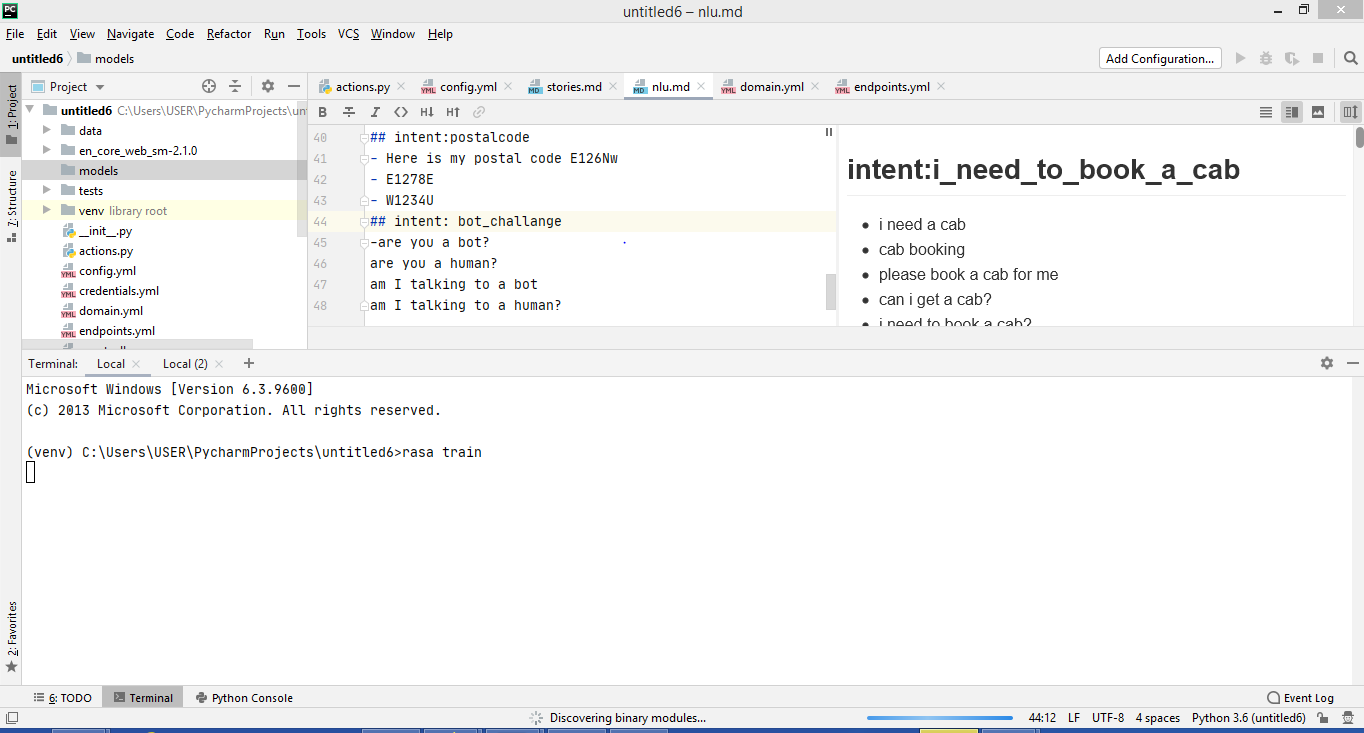
1. Rasa train
2. Rasa shell –debug

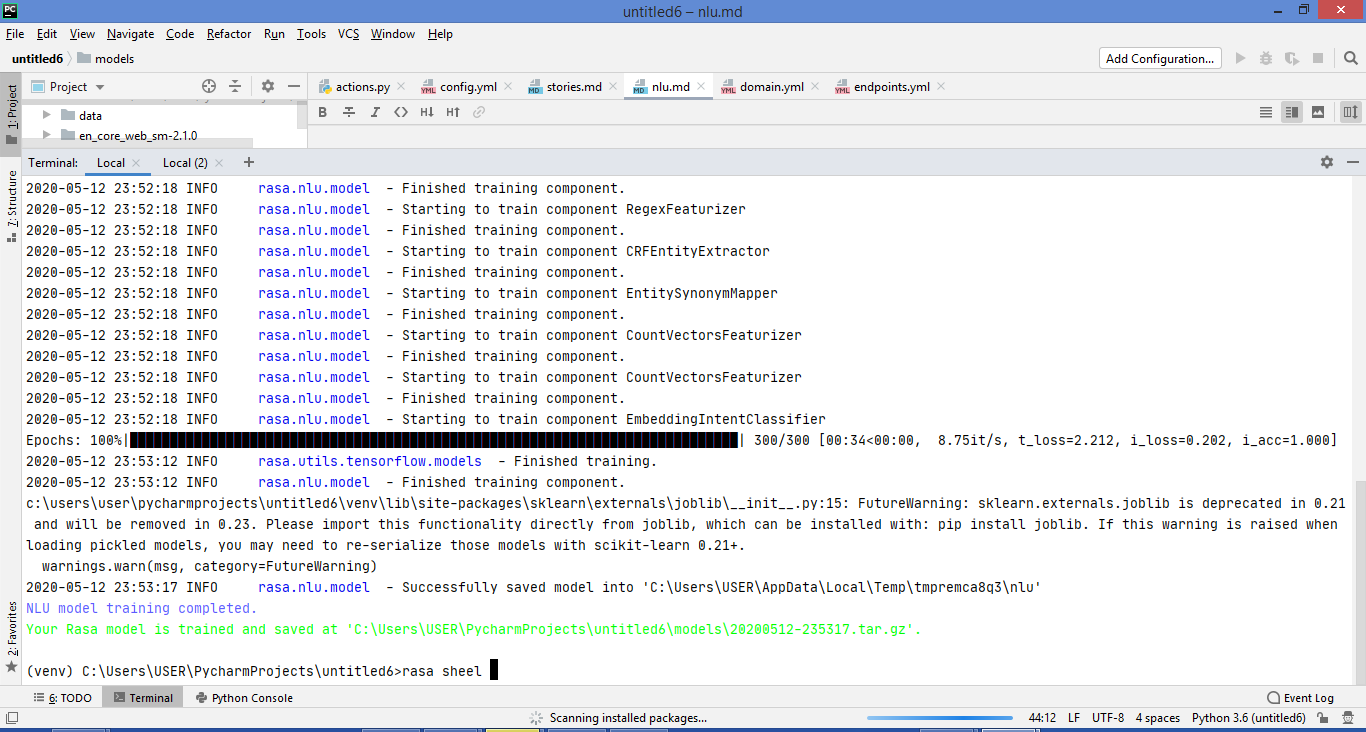
To run the chatbot in browser

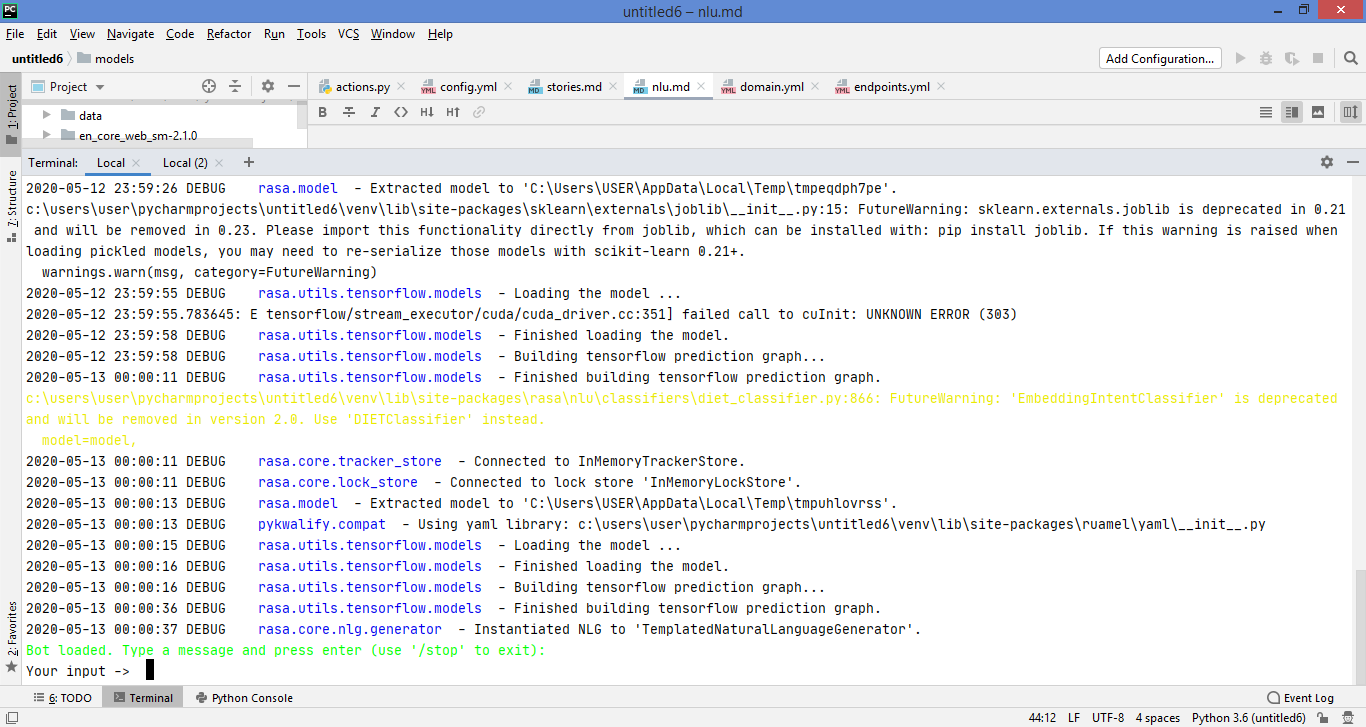
1. Rasa train
2. Rasa x

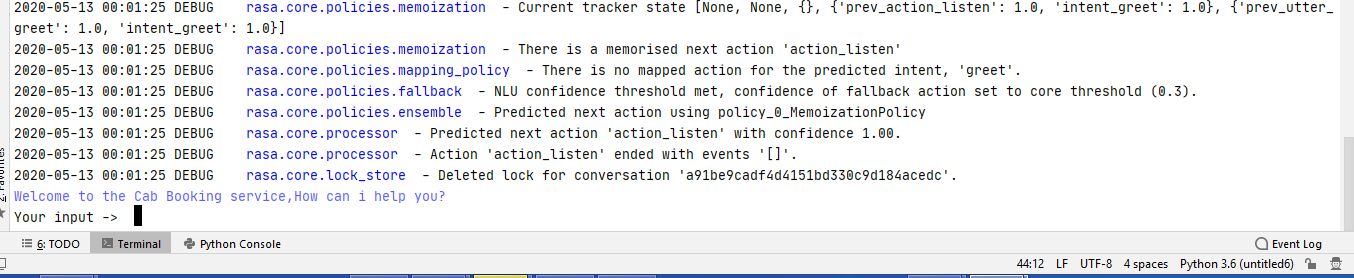
We have created a chatbot that helps user to book or cancel a cab and it majorly includes the three main modules:

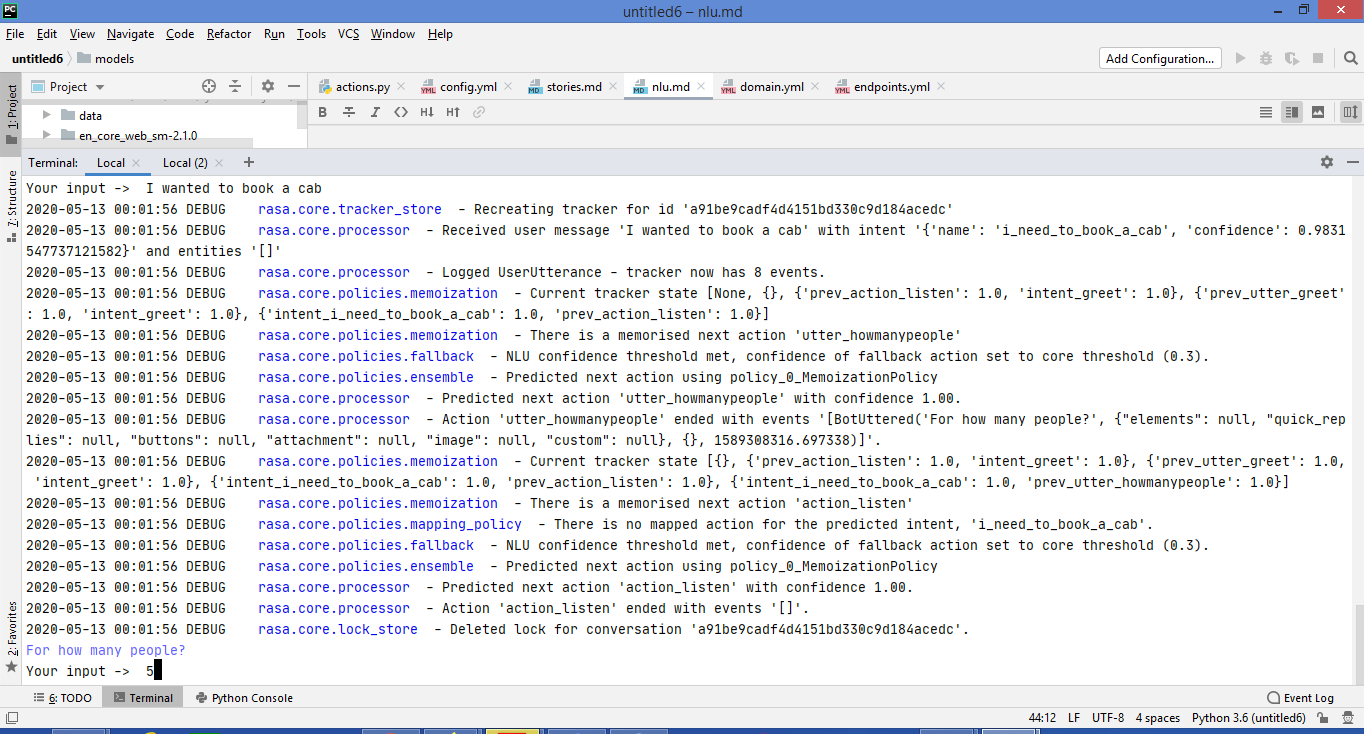
* 1. Booking a cab: Below screenshots states the execution part for booking a cab. This is shown in the debugger mode:

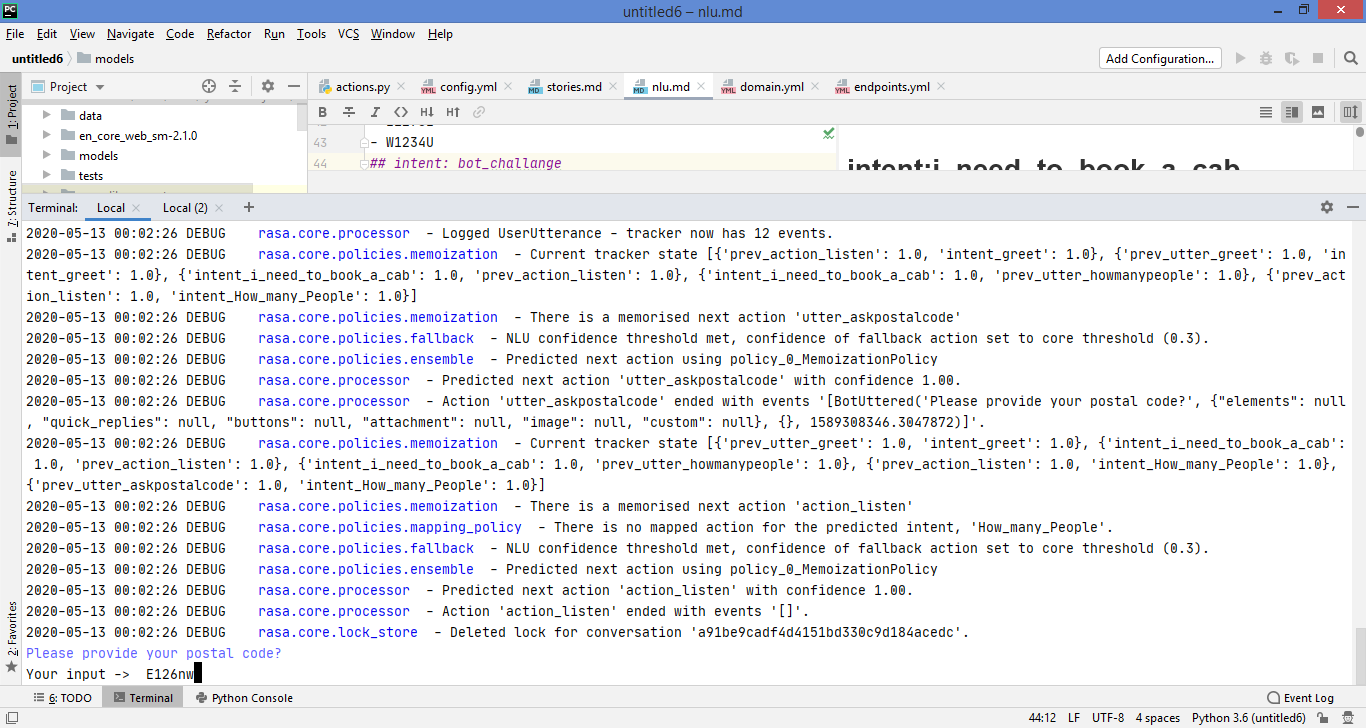


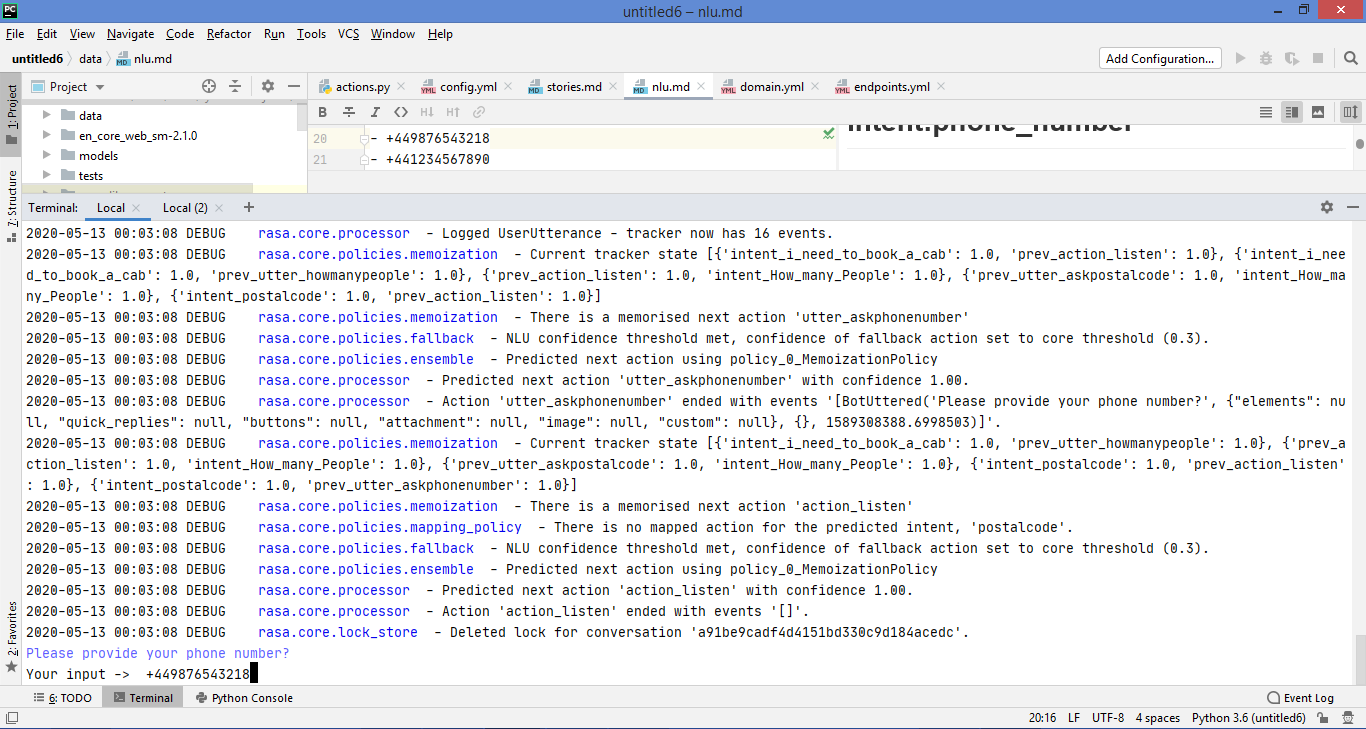


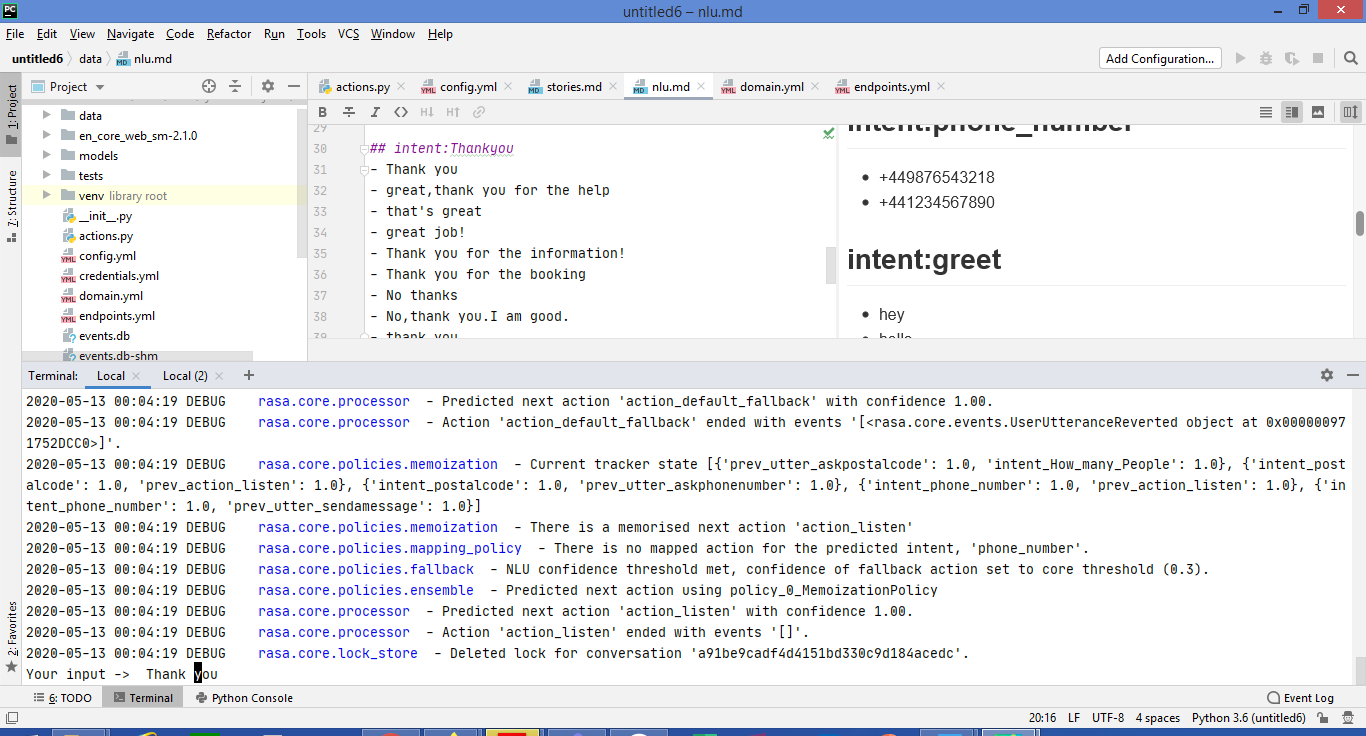




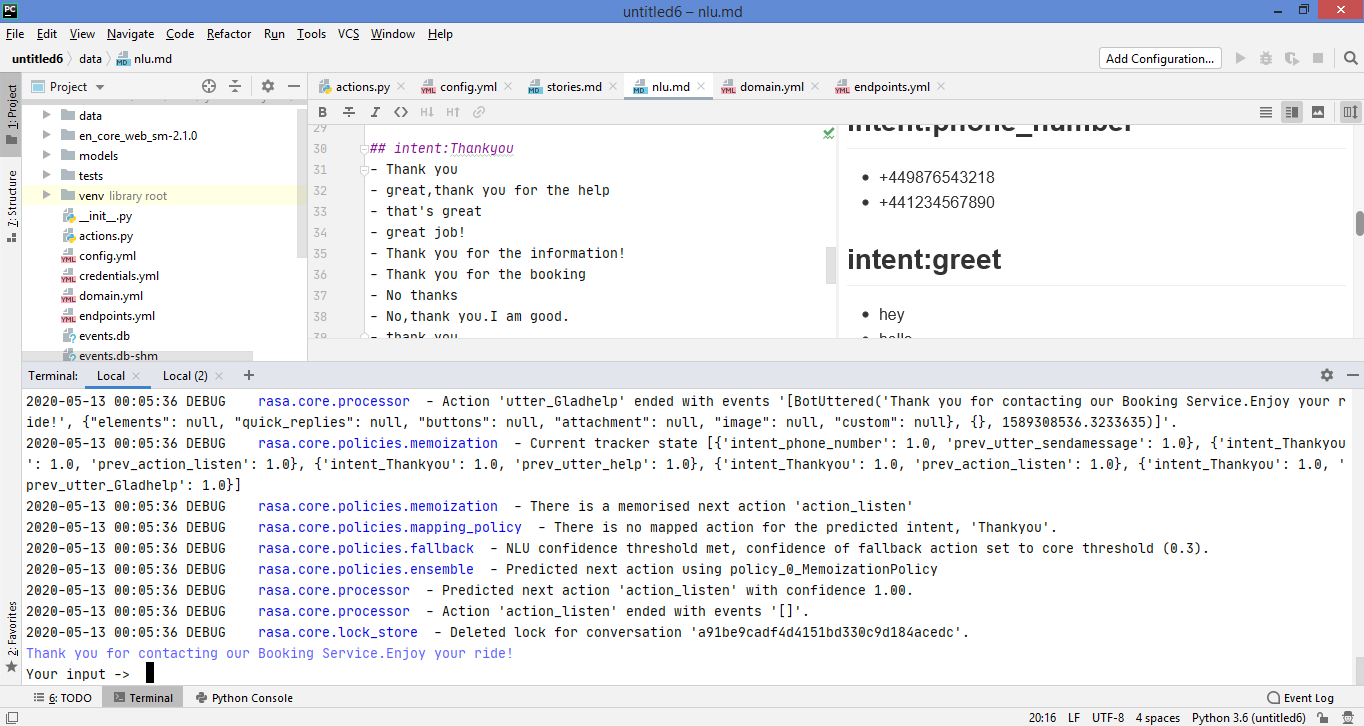




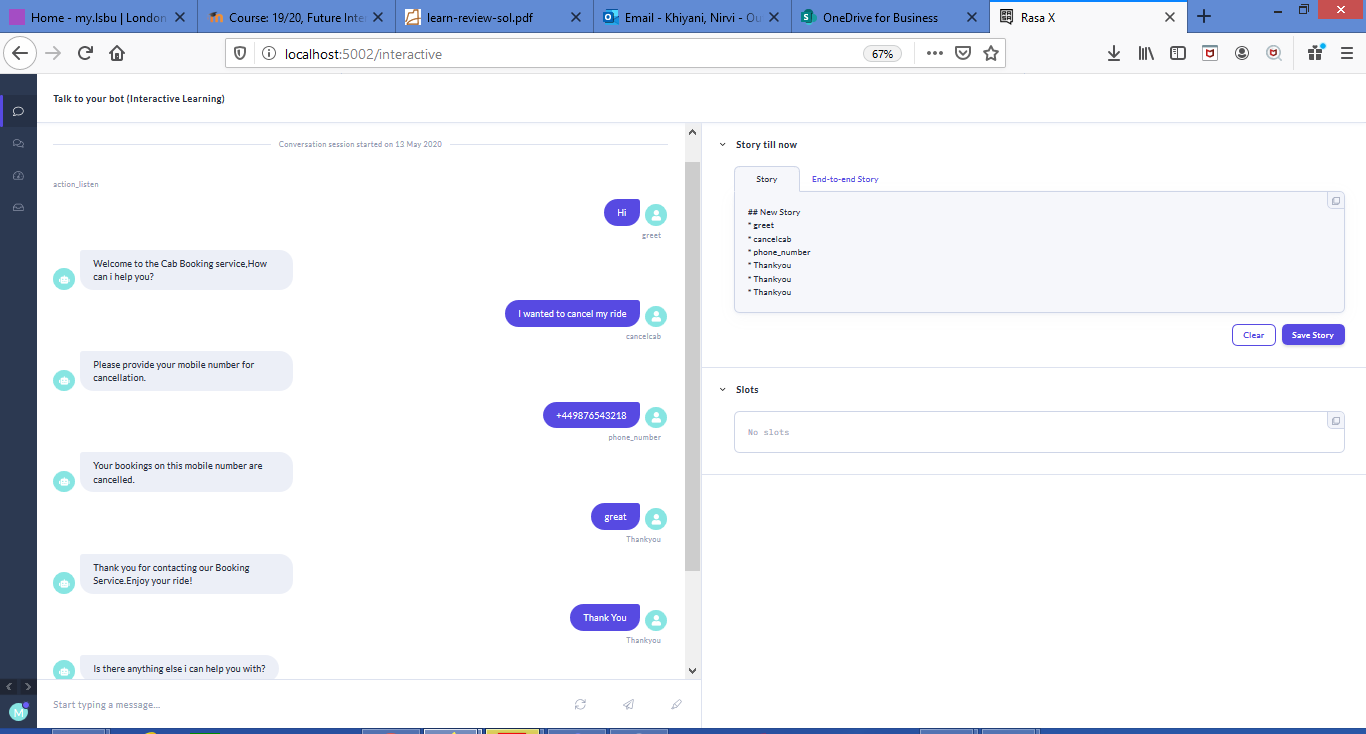


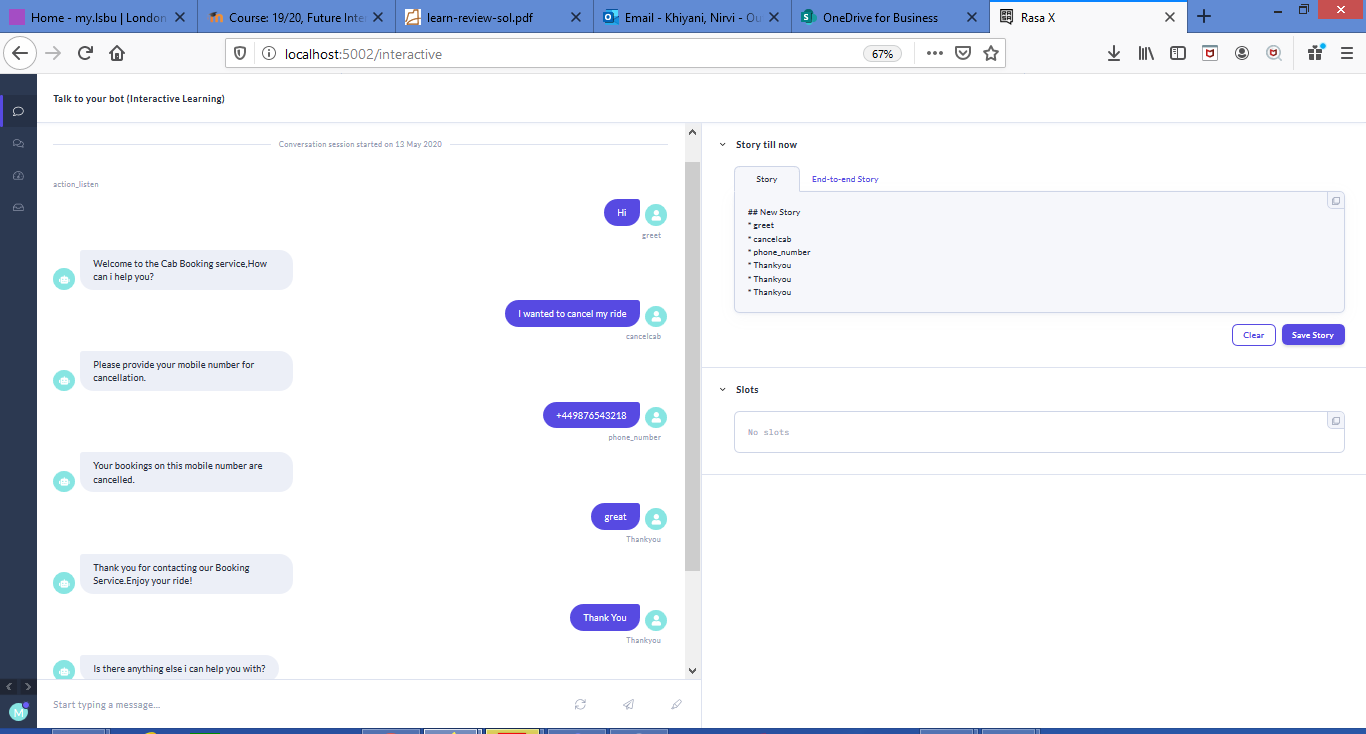






**b)Cancelling a cab:** Below screenshots states the execution part for booking a cab. This is shown in the browser window:

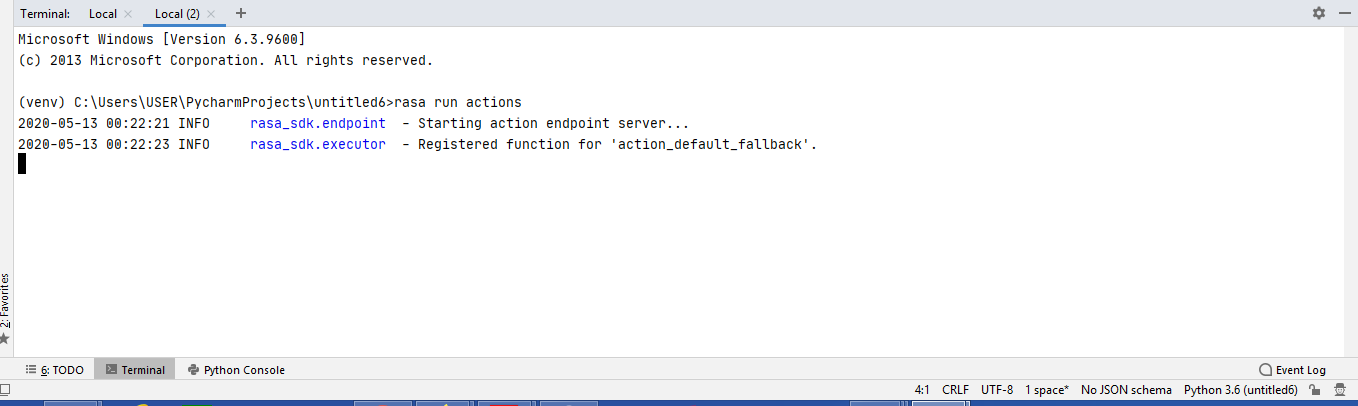




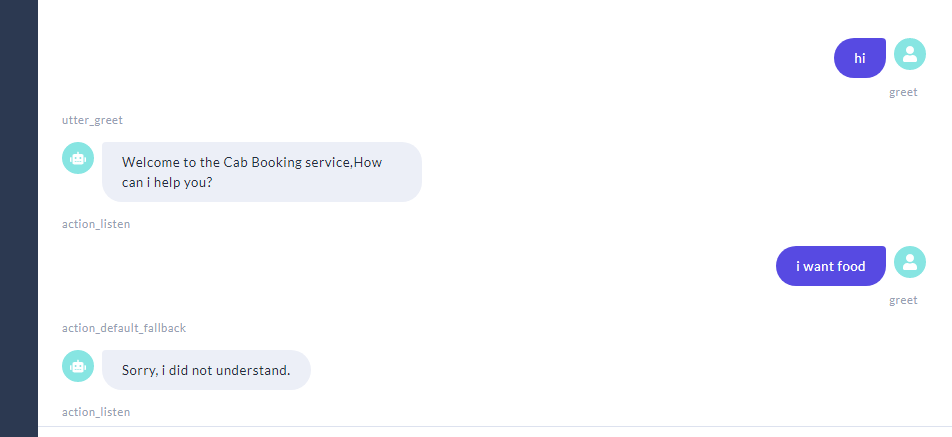
1. **Fallback event:**

There should be error handling for all the front and back end application and hence, it requires for Chatbot also. Here, in this Rasa and chatbot terms error handling is termed as Fallback action where, the chatbot has some default set of answers when the user asks the question which is either not in scope or out of the box. The policy of fallback action is termed as fallback policy and is executed when confidence of intent recognition is below nlu\_threshold or either prediction of dialogue policy has a confidence which is greater than core\_threshold.

The default action in Rasa core is action\_default\_fallback then sends utter\_default response to the user. Also, we need to include the utter\_default in the domain file and hence, we have included that in our domain file to handle the fallback action. In order to run a fallback event and to provide the answers for out of the box questions in the debugger mode the “rasa run actions” command should be running simultaneously in the next terminal window. It is stated below:



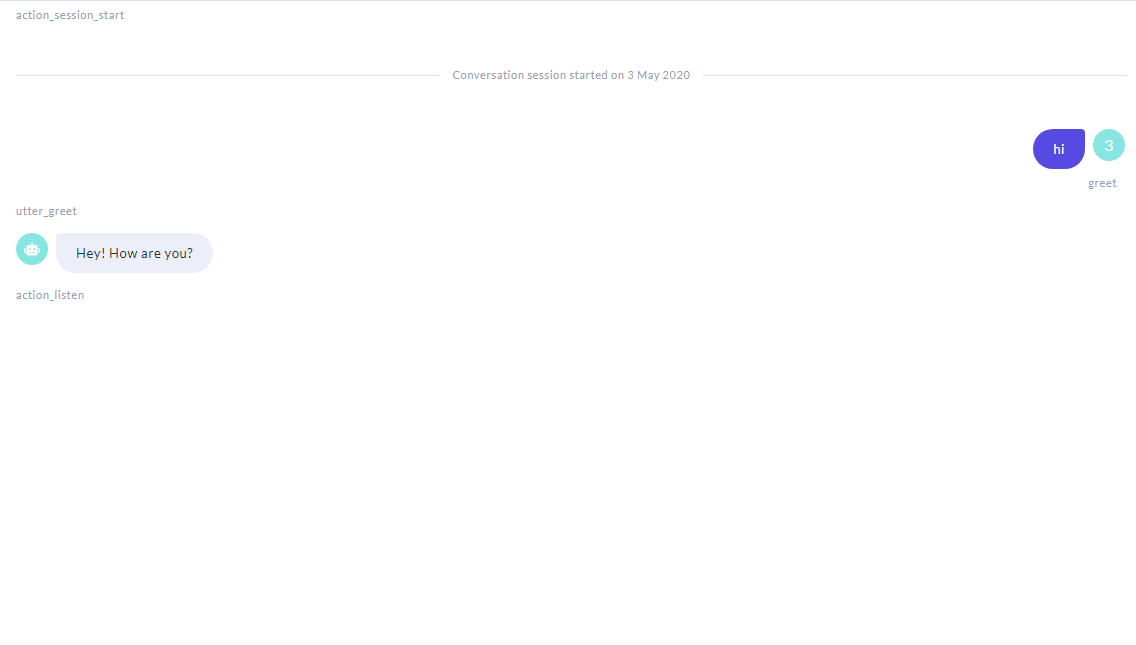
Below screenshots states the execution part for booking a cab. This is shown in the browser window.



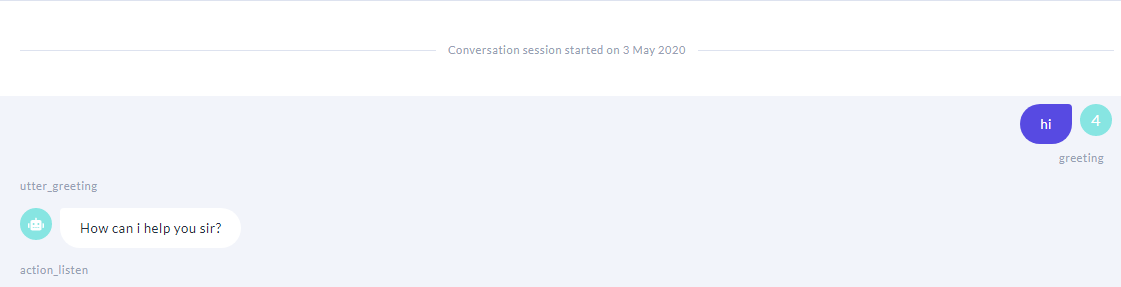
**7.2 Execution errors:**

While executing there were several errors that were occurred, and they are stated as under along with the screenshot:

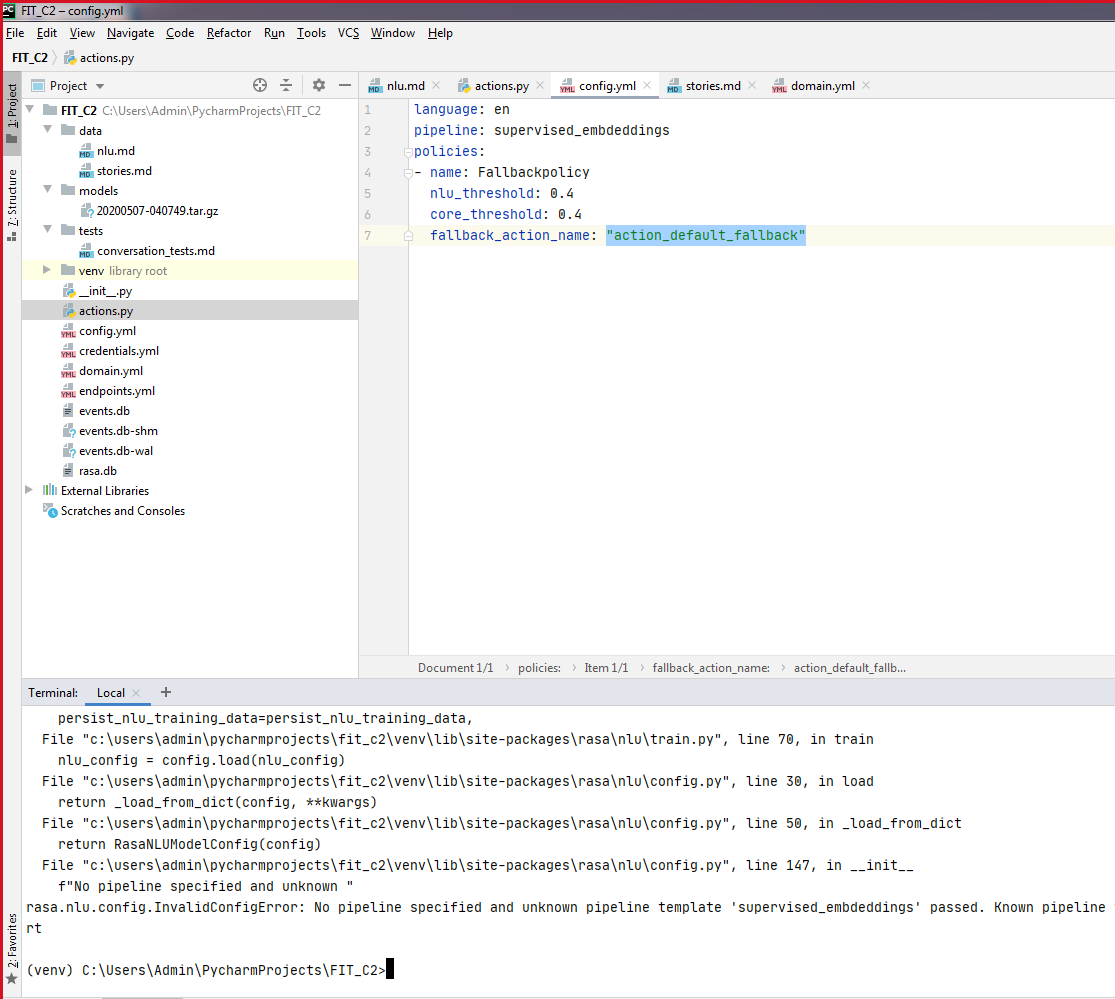
Even though intents were changed, it was still taking old intents only rasa x

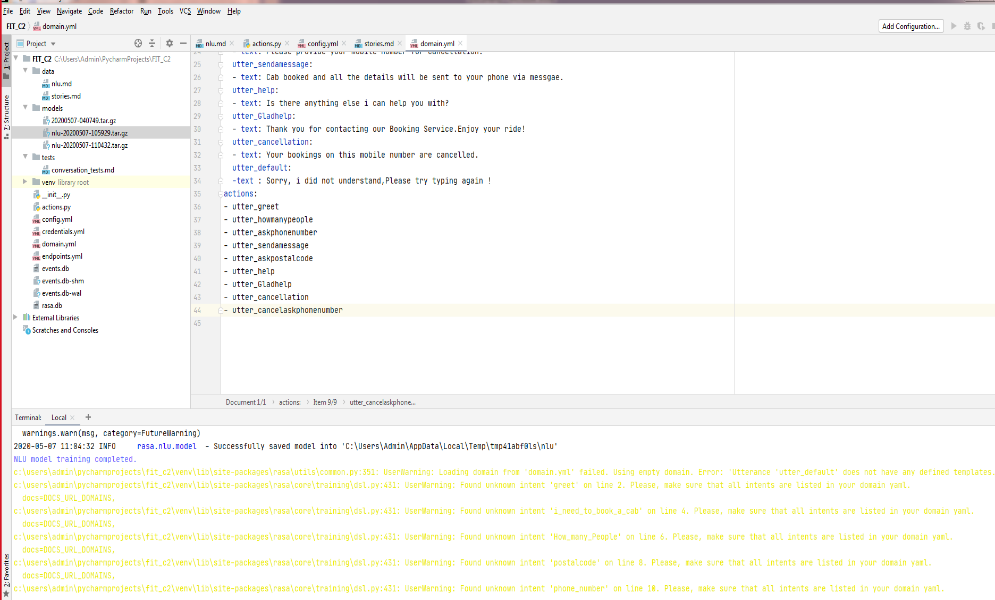


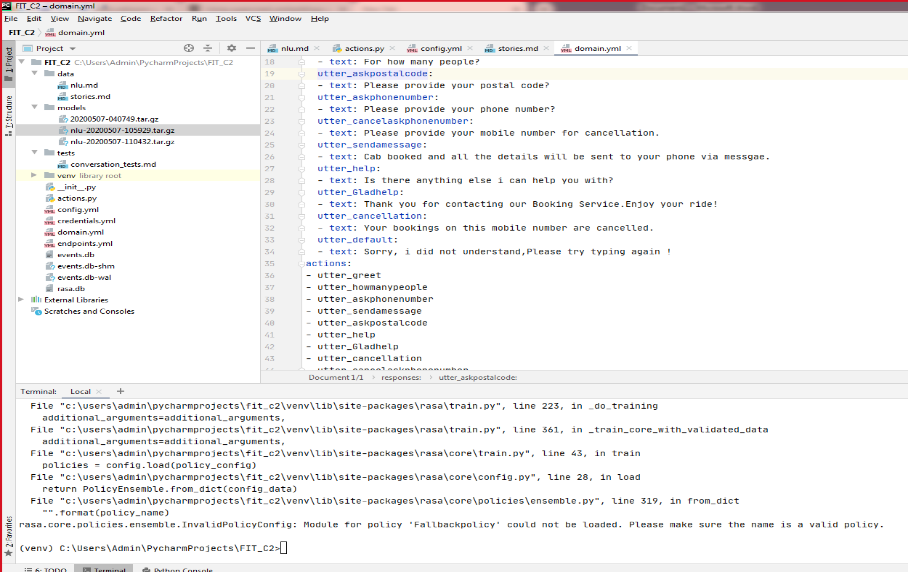
This was resolved bydeleting all by default models created by rasa and wrote the new intents



Lastly, while executing the fallback actions i.e; default actions we were getting multiple errors as stated below:









Tried changing all the nlu\_threshold and core\_threshold and tried updating, several policies and pipeline, but nothing worker. At the end, when we ran “rasa run actions” simultaneously in other terminal window then at the end this issue was resolved and finally we were able to run fallback action

**8.Performance and Accuracy:**

The performance of a chatbot can be measured by using RSA NLU pipeline to extract intents and entities. For this, a trainer from rasa nlu is imported which trains the data and then pass it to interpreter. Accuracy of intent is defined as the number of intent that bot responded correctly. The accuracy of the intent is given by:

Intent Accuracy: total correct recognized intent/ total intents.

On the other hand, entity accuracy is defined as number of entities that are extracted correctly divided by total number of entities

Entity Accuracy: correctly extracted entities/total number of entities

Entity Integrity: Recognized events/total entities

Sentence Integrity: total extracted lines/total lines.

Overall, my bot response time was very less which states its performance was good, having intent accuracy as ’0.99’ and entity accuracy as ‘1.0’ and hence the accuracy was 1.0 from Rasa NLU method having integrity entity as 0.95 and integrity of sentence as 0.92.

**9. Conclusion:**

NLU is the crucial component of chatbot systems. This report includes the chatbot model that helps users to book a cab or cancel a cab by using RASA NLU principles that is introduced for chatbot system. It integrates methods like neural network and RASA NLU and makes implementation of system which is entity extraction based which takes place after the recognition of intent. This reports majorly highlights the errors that were encountered while installing and building a transport chatbot and how those error were troubleshooted. In the end of the report, the results of the performance tests are provided which is done by using Rasa NLU method.

**10. References:**

[1] Cambria E, White B 2014 Jumping NLP curves: a review of natural language processing research J. IEEE Computational Intelligence Magazine 9(2) p 48-57.

[2] Kamper H and Ruder 2018, October 1 A Review of the Neural History of Natural Language Processing Aylien Retrieved March 18, 2019, from recent-history-of-natural-languageprocessing/

[3] Bocklisch T, Faulkner J, Pawlowski N and Nichol A 2017 Rasa: Open Source Language Understanding and Dialogue Management arXiv: Computation and Language CCEAI 2020 IOP Conf. Series: Journal of Physics: Conf. Series 1487 (2020) 012014 IOP Publishing doi:10.1088/1742-6596/1487/1/012014 9

[4] Desot T, Raimondo S, Mishakova A, Portet F and Vacher M 2018, September Towards a French Smart-Home Voice Command Corpus: Design and NLU Experiments C. text speech and dialog