Chatbot for E-Commerce Assistance: based on RASA

M.Mamatha[1], C.Sudha[2]

[1,2]Department of Computer ScienceEngineering,Mahatma Gandhi Institute of Technology, India

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May2021

Abstract—Whenever a customer using an Ecommerce sites like Amazon, Flipkart etc, he may face issues which may trouble him. It takes time for the customer support to resolve the customer issues since billions of people are using those platforms and reporting issues regularly. So this makes some congestion on the customer support side, so that they could not react so quickly. One more issues are buying the products by searching from different platforms. This may also takes too much time. So here, adding a bot to the platform which can understand the human language and generate an appropriate response. This bot will be useful for filtering the products from whatever the ecommerce sites it has been incorporated it with(here the own site developed, which runs in local server as other ecommerce api procurement is taking much time than expected) ans also replying to some of the issues before they get to the customer call center.

Keywords—AIML, E-Commerce Assitance, Intelligent Chatbot, Smart Chatbot

I. INTRODUCTION

Ecommerce platforms are a boon now a days. Billions of people were using different ecommerce platforms to buy their essentials. It may be clothing[1], electronics, medicine. It may rise further in the future. With this, the issues pertaining to their orders or any payment were also raising and may be even difficult to get answer to the queries. Whenever we want to purchase any product, we need to search various platforms to choose which is the best one among all. This is time consuming, We have no platform where we can find all the products available at one place. This is frustrating for most of the people. Here we are building a piece of program called a Chatbot[4] where the user will be interacting with this piece of program to report some of the frequently occurring issues in the ecommerce platforms and get answers to their queries. Also they can filter the products from this bot itself. They no longer needed to search them on all the platforms. Here[2] we are building this ChatBot using Rasa Framework

In essence, a Chatbot responds to simple chat by processing user requests based on a collection of questions specified in the knowledge base. In order to respond smarter, quicker, and more accurately, a chatbot is currently being created. Data parsing, pattern matching, and data crawling are only a few of the processes used in the chatbot framework. Any experiments do not use the whole procedure in order to minimize computing time. According to the study, the chatbot system only employs two processes: sorting and data crawling. Based on the outcome of the 1500 queries, only 1200 can be replied correctly, or about 80% accuracy. Other researchers only recommended pattern matching and data creeping, with a 95 percent accuracy rate and response times ranging from 7.5 seconds to 48 seconds.

Grammar-based data parsing is needed for efficient Chatbot applications in order for the user to comprehend the intended sentence by defining phrases that are suited to the complexities of the grammar used.

Efficient Chatbot applications require grammar-based data parsing to assist understand the intended sentence by the user, by describing phrases that are adapted to the complexity of the grammar used. Some parsing stages[2] are administered with several methods, namely case folding, tokenizing, filtering, and stemming. But many problems occur during the pattern matching process because there are various rules that haven't been standardized thanks to inappropriate data parsing processes, an honest pattern matching process are often established if the applied pattern can sequentially extract information that's useful for analyzing the relevant text, the way is by eliminating irrelevant information and selecting during the parsing process.

Crawling data is that the last process wont to search data during a database that matches the pattern matching results. RASA is actually a framework to create very friendly user interacting chatbot. Whatever the processes that have mentioned above like tokenizing, stemming all this will be done internally by the rasa framework. What we have to do is basically use the modules it provides to its effect. Even machine and deep learning models are need not be created from scratch. Rasa it self has builted models which are in abstraction. Only the commands like rasa train will train the model and it will be ready for use.

II. RELATED RESEARCH

Nagdev et al use AIML and A.L.I.C.E to conduct a monotonous query/answer procedure with the intention of eliminating semantic complexity in communication and achieving the appropriate output. In the experiments, FUTURE bots and NEST bots with huge datasets of over 50,000 responses were used. This visit was recorded in the form of a text containing 1500 questions[2], but only 1200 of them could be answered correctly, resulting in an accuracy of 80%. Keywords from objects must be selected from a list of documents that can be contained in the database.

Mhatre et al, on the other hand, used AIML and pattern matching in machine learning to approach web-based bots that stimulate by sharing knowledge that occurs via e-mail communications through analyzing AIML files that involve the use of language processing algorithms such as pattern matching. According to tests performed with the Donna Collaborative Bot on 700 tweets, the timing response ranges from 7.5 seconds to 48 seconds, with a poor response of 5%, an outstanding response of 35%, and a response that is almost equal to requests of 60%. In terms of the efficiency of the system to respond within a specified time period, achieving an overall efficiency of 70% because when input increases, most of the system efficiency decreases which is a limitation for the system in responding has taken a long time to respond to client requests.

Pradana et al. used AIML and ELIZA in SamBot by using a log-based information management module and received 132 points and a score of 26.4 points on the Turing Test or Loebner Prize chatbot around the world. There were 100 questions sent by internet users for this study, but 30 of them were unacceptable to answer to. According to this report, information is restricted to bot master awareness, allowing the chatbot to respond inappropriately to feedback and produce random responses, interfering with users' ability to communicate with a chatbot.

Tufai et al used an A.L.I.C.E algorithm to model communications and AIML to shape text-based answers by pattern matching messages against the questions provided to come up with a successful answer. However, since the knowledge base that is held is very small, this study cannot guarantee details about how much time is required in response. As a result, in order to find relevant facts, the conversation volume will rise, potentially leading to inconsistencies in user connectivity with the chatbot.

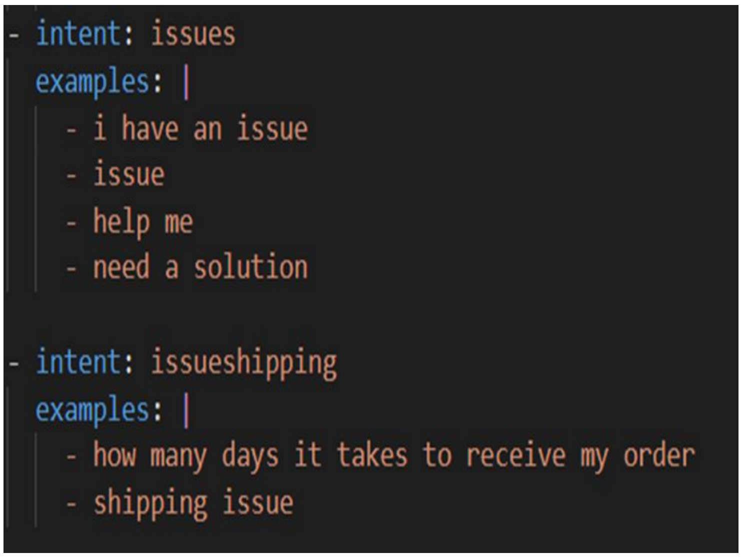
III. PROPOSED METHOD

A. Rasa Modules

A raw chatbot build on top of deep learning and machine learning techniques basically uses any of the frameworks like tensor flow etc. They need to be modeled from scratch. But using Rasa we simply use the modules provided by rasa to train the inbuild model with the input data where the intent from the user can be get and also customizing the output for the user intent.

1.Rasa Nlu

Rasa NLU is an important module or file where a set of intents are defined so that the intent drawn from the user input text will be matched with any of the intents.



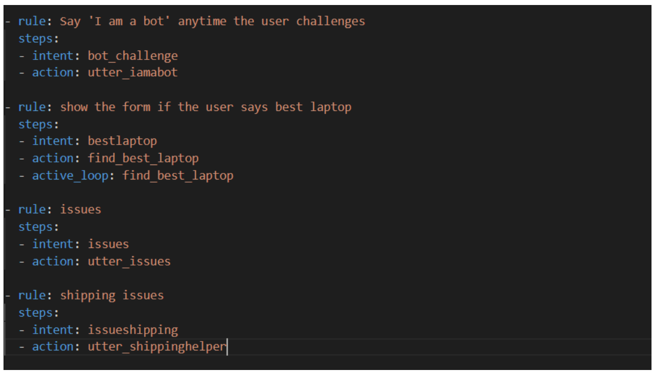
```
- intent: issues
  examples: |
    - i have an issue
    - issue
    - help me
    - need a solution

- intent: issueshipping
  examples: |
    - how many days it takes to receive my order
    - shipping issue
```

Fig. 2. Examples showing the intent in various ways

From the above figure we can see there are examples with number of text sentences and there are intents for each example

2. Rasa Rules

RASA RULES are nothing but whenever there is a input text that has an intent which matches with intent in the rules domain of the RASA RULES module, then a specific action will be performed.



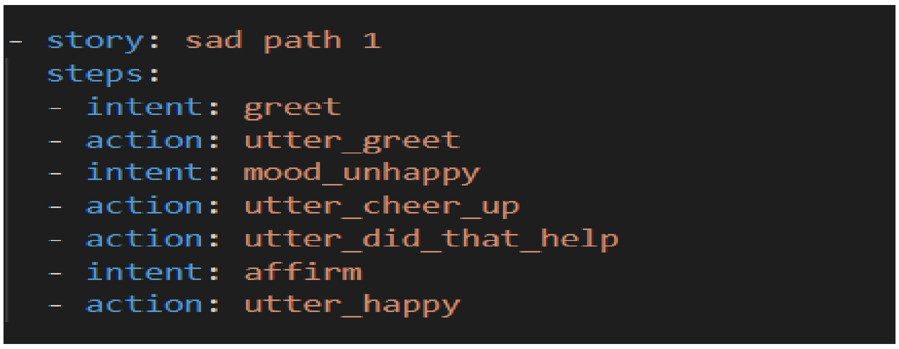
```
- rule: Say 'I am a bot' anytime the user challenges
  steps:
  - intent: bot_challenge
  - action: utter_iamabot

- rule: show the form if the user says best laptop
  steps:
  - intent: bestlaptop
  - action: find_best_laptop
  - active_loop: find_best_laptop

- rule: issues
  steps:
  - intent: issues
  - action: utter_issues

- rule: shipping issues
  steps:
  - intent: issueshipping
  - action: utter_shippinghelper
```

Fig.3 shows how to write a rule to call an action for an intent of user

From the above figure we can see for a specified intent, there is an action. From figure 3.2 if you say “I have an issue” bot will draw intent as “issues”. From figure 3.3 action for intent issues is utter_issues which a template that is rendered to the user as a response. We will discuss this in the coming chapters.

3.Rasa stories

Rasa stories are nothing but a flow of set of intents and responses in a conversation which takes place one after another. From the below figure, you can figure it out.



```
- story: sad path 1
  steps:
  - intent: greet
  - action: utter_greet
  - intent: mood_unhappy
  - action: utter_cheer_up
  - action: utter_did_that_help
  - intent: affirm
  - action: utter_happy
```

Fig.4 story build up with sequence of intents and actions

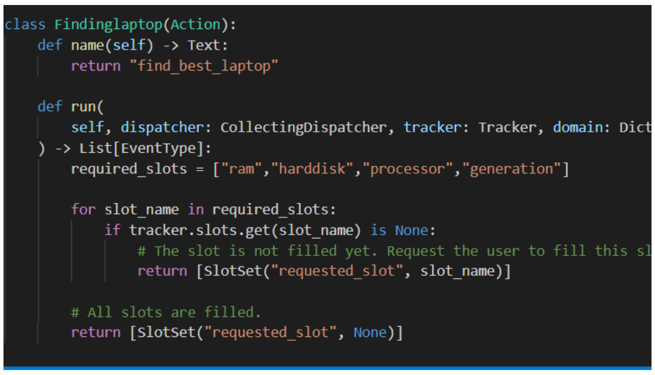
From the about figure, It is observed that for a story, there are steps which takes place one after the other. These steps are basically intents and responses. In the figure it can be seen that whenever a user inputs a text with greet intent, a response utter_greet is generated. This is one set of input and response. This goes on with each set in the steps that is one at a time and that too one after other in sequential order in a conversation. Assume it like a conversation where intent and action are two different persons talking with each other i.e one dialogue after another.

4.Domain.yml

From figures 2,3,4 whatever the actions that are there are nothing but responses which are defined in this module. From the below figure, you can see the responses defined. Lets see an example. For input text “I have an issue” from user intent of “issues” will be matched from figure 2.So for this intent there are both rules and stories defined. It can pick any action defined to that intent. Here from figure 3 utter_issues is an action which is defined in domain.yml with response as “kindly let me know your issue” which will be sent as response to the user.

5.Custom Actions

Custom actions can be used to generate custom responses to the user intent. This can be done by using the module actions.py in actions folder. You can customize or create new class presented in it. So after getting the intent from the user, the next step is to generate a response. The response can be from domail.yml where an action has a definition inside it. But on the other hand we define an action in actions.py for custom actions by returning a action name from the function. That returned name from the function can treated as action name. Whenever you put this action name inside an action just like you did above, that function which is return the name will be called. A detailed figure to understand is as follows

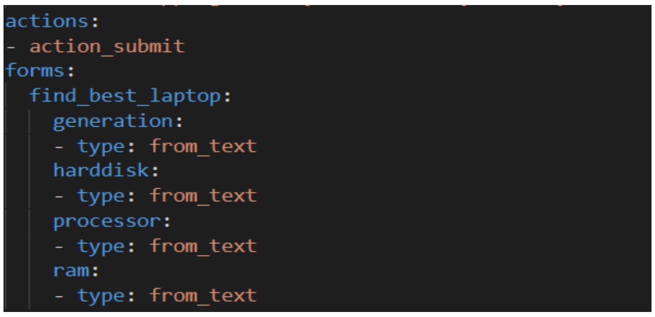


```
class Findinglaptop(Action):
    def name(self) -> Text:
        return "find_best_laptop"

    def run(
        self, dispatcher: CollectingDispatcher, tracker: Tracker, domain: Dict
    ) -> List[EventType]:
        required_slots = ["ram","harddisk","processor","generation"]

        for slot_name in required_slots:
            if tracker.slots.get(slot_name) is None:
                # The slot is not filled yet. Request the user to fill this sl
                return [SlotSet("requested_slot", slot_name)]

        # All slots are filled.
        return [SlotSet("requested_slot", None)]
```

Fig.5 shows the custom action file for generating custom responses

From the above figure, you can see there is class called Finding Laptop, inside it there were two functions namely name and run. The name function returns the action name and run function executes main theme you want to intent is best laptop, then there is an action name called find_best_laptop.so you can now correlate that action name with the return of the name function in the class which is shown in the above figure.

Now run method is executed itself without explicitly invoking it. When you want to collect data from the user(here I have collected laptop configurations from the user),you need to create a form in the domain. Yml, which can be shown below.



```
actions:
- action_submit
forms:
  find_best_laptop:
    generation:
    - type: from_text
    harddisk:
    - type: from_text
    processor:
    - type: from_text
    ram:
    - type: from_text
```

Fig.6 showing how to create a form

Before creating a form, firstly you need to create slots. Slots are nothing but like variables where you store you input data.

After this, you need to create a rule in rules.yml to submit the form.

Now we can filter the products from the user filled slots, which are then sent as a request using rest api of a site from the custom action. The results are then send back to the user

IV. BUILDING CHATBOT SYSTEM

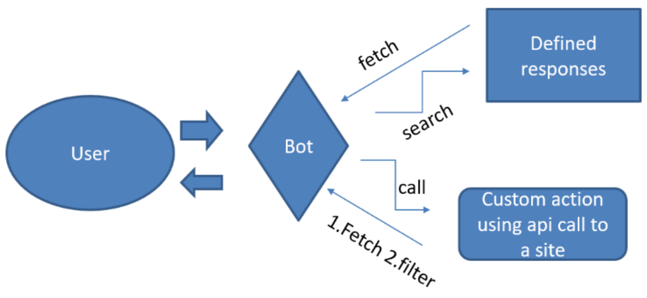


Fig.7 Chatbot Model

Now using the respective modules we have discussed above are used to build the chatbot which has the above structure represented. Here User may interact with the Chatbot in any of the two forms namely command line and the user interface. Whenever a user type something and press enter, the message or the input will be taken by the bot and the user intent is drawn from the input text. Based on the intent Bot will decide whether the result for the input query is required any external api call.

After verifying, Bot may fetch the result either from its defined response which are in the domain.yml or by making an api call to an external site or database. The results thus obtained are given to the user through the same interface(command line) or graphical user interface

V.Evaluation and testing

Features to be tested

1. Intent recognition and replay back with appropriate response
2. Solution to the shipping issue
3. Slot filling of laptop specifications
4. Api connection and request data from a site
5. Filter the laptops when there are many laptops in the site with required parameters

All the test cases were successful with the results are appropriate enough with our expected ones.

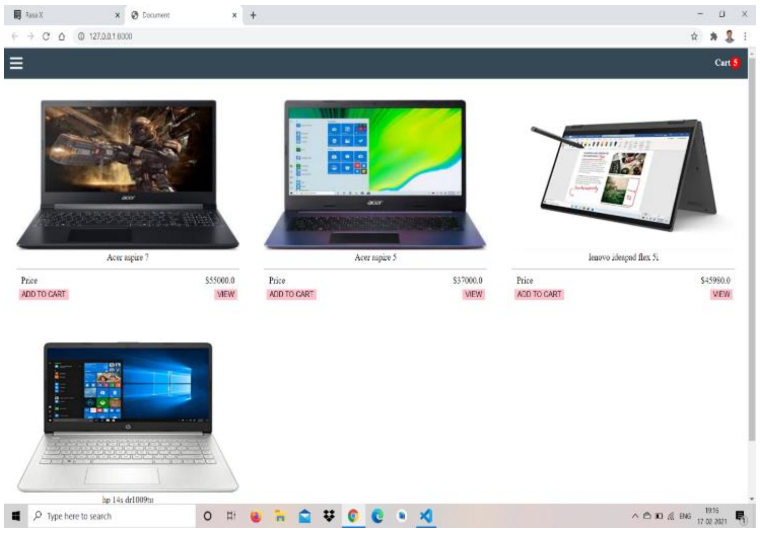
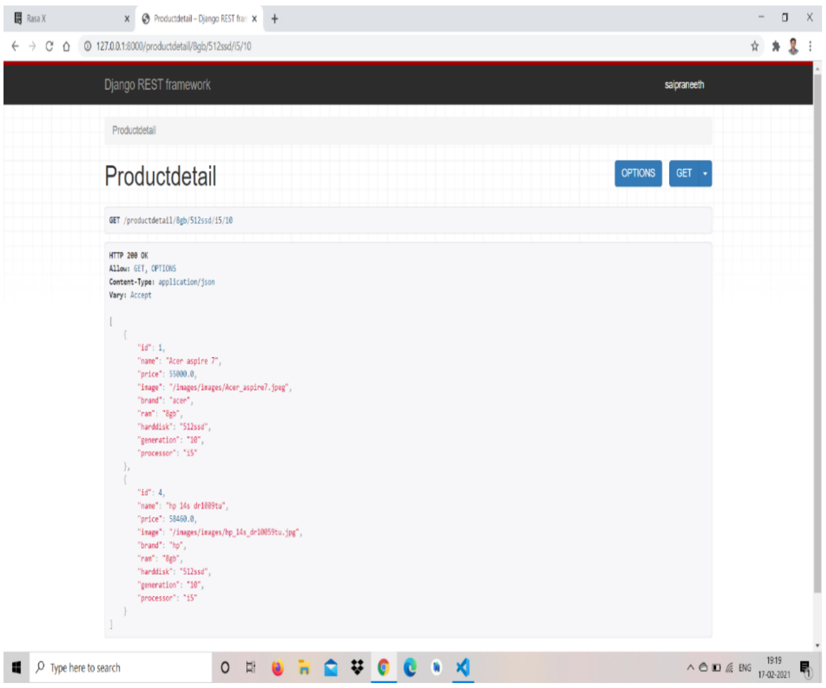


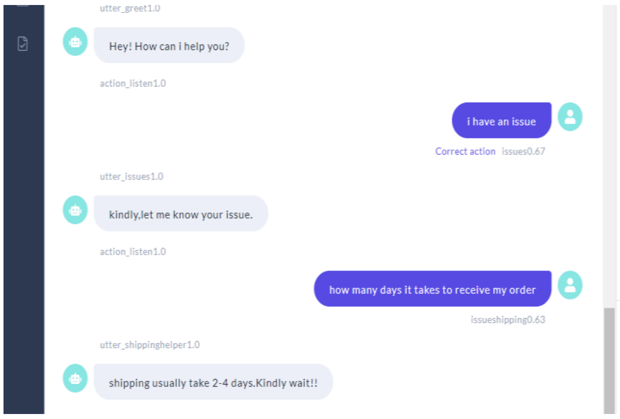
Fig.8 site where products are filtered



Productdetail

```
GET /productdetail/8gb/512ssd/i5/10

HTTP 200 OK
Allow: GET, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "name": "Acer aspire 7",
        "price": 55000.0,
        "image": "/images/images/Acer_aspire7.jpeg",
        "brand": "acer",
        "ram": "8gb",
        "harddisk": "512ssd",
        "generation": "10",
        "processor": "i5"
    },
    {
        "id": 4,
        "name": "hp 14s dr1009tu",
        "price": 58460.0,
        "image": "/images/images/hp_14s_dr10059tu.jpg",
        "brand": "hp",
        "ram": "8gb",
        "harddisk": "512ssd",
        "generation": "10",
        "processor": "i5"
    }
]
```

Fig.9 json response of an api call



utter_greet1.0

Hey! How can i help you?

action_listen1.0

i have an issue

Correct action issues0.67

utter_issues1.0

kindly,let me know your issue.

action_listen1.0

how many days it takes to receive my order

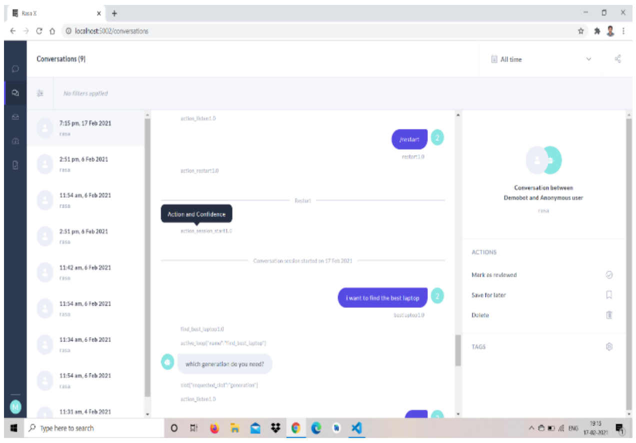
issueshipping0.63

utter_shippinghelper1.0

shipping usually take 2-4 days.Kindly wait!!

Fig.10 showing the conversation of bot shipping issue

From the above figure, user has started a conversation. In the meanwhile, he is having an issue with his order and the correctly grasp the intent of the user, which is about an issue with his order and the bot replied accordingly.



localhost:5002/conversations

Conversations (9)

No filters applied

7:15 pm, 17 Feb 2021 rasa
2:51 pm, 6 Feb 2021 rasa
11:54 am, 6 Feb 2021 rasa
2:51 pm, 6 Feb 2021 rasa
11:42 am, 6 Feb 2021 rasa
11:54 am, 6 Feb 2021 rasa
11:34 am, 6 Feb 2021 rasa
11:54 am, 6 Feb 2021 rasa
11:31 am, 4 Feb 2021

action_listen1.0

/restart

restart1.0

action_restart1.0

Restart

action_session_start1.0

Conversation session started on 17 Feb 2021

i want to find the best laptop

bestlaptop1.0

find_best_laptop1.0

active_loop{"name":"find_best_laptop"}

which generation do you need?

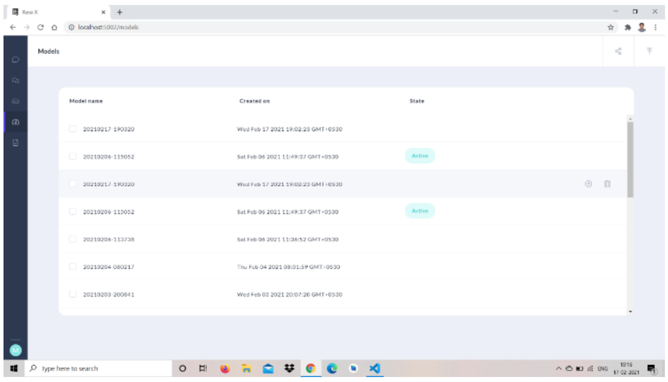
slot{"requested_slot":"generation"}

action_listen1.0

Conversation between Demobot and Anonymous user

Fig.11 showing the previous conversations.

This chatbot can store you last conversations and you can view them at any time. From the left side of the figure, you can see the previous conversations with their timings.



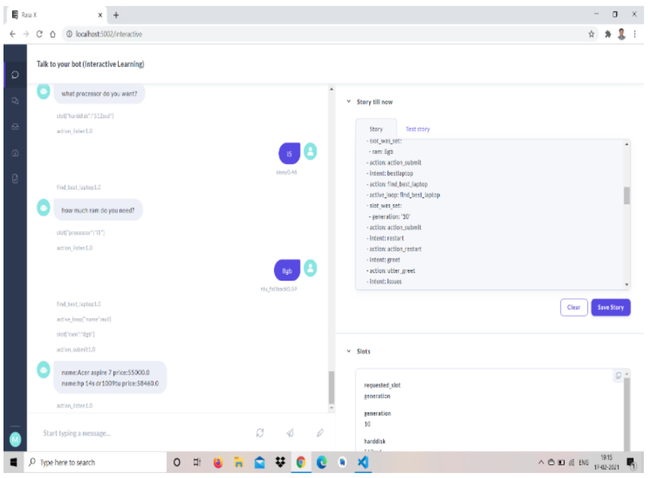
localhost:5002/models

Models

Model name	Created on	State
20210217-190320	Wed Feb 17 2021 19:02:23 GMT+0530	
20210206-115052	Sat Feb 06 2021 11:49:37 GMT+0530	Active
20210217-190320	Wed Feb 17 2021 19:02:23 GMT+0530	
20210206-115052	Sat Feb 06 2021 11:49:37 GMT+0530	Active
20210206-113736	Sat Feb 06 2021 11:36:52 GMT+0530	
20210204-080217	Thu Feb 04 2021 08:01:59 GMT+0530	
20210203-200841	Wed Feb 03 2021 20:07:20 GMT+0530	

Fig. 12 showing the models trained correctly and the current active one

From the above figure it is observed that the different models trained at times have shown. You can make any model active and start the conversation with the ChatBot right click on the model and then you can see the activate option.


Rasa X
localhost:5002/interactive
Talk to your bot (Interactive Learning)
Story till now
Story
Test story
Clear
Save Story
Slots
Start typing a message...
Type here to search

what processor do you want?

slot{"harddisk":"512ssd"}

action_listen1.0

i5

deny0.46

find_best_laptop1.0

how much ram do you need?

slot{"processor":"i5"}

action_listen1.0

8gb

nlu_fallback0.59

find_best_laptop1.0

active_loop{"name":null}

slot{"ram":"8gb"}

action_submit1.0

name:Acer aspire 7 price:55000.0
name:hp 14s dr1009tu price:58460.0

action_listen1.0

```
- slot_was_set:
  - ram: 8gb
- action: action_submit
- intent: bestlaptop
- action: find_best_laptop
- active_loop: find_best_laptop
- slot_was_set:
  - generation: '10'
- action: action_submit
- intent: restart
- action: action_restart
- intent: greet
- action: utter_greet
- intent: issues
```

requested_slot
generation

generation
10

harddisk

19:15
17-02-2021

Fig 13 showing the filtered laptops

In the above figure, user started a conversation saying the laptop specifications. These specifications are appended to an api, the data is rendered as json which is shown in figure4.2. The data thus acquired is rendered as a template. You can see it in the Figure.

VII.CONCLUSION

Here we have successfully created a ChatBot using Rasa. The main Function of our ChatBot is to address the issues relating shipping in our site and also the main aim of our ChatBot is to filter the products from the site. Here I have filtered only laptops. Rasa is an essential tool or Framework to build a ChatBot. Rasa X is an extended version of Rasa which provides GUI where user can interact with it. The main advantage of Rasa chatbot is basically the easiness and customization of a chatbot without having in depth knowledge in deep neural networks and machine learning. This project can be extended to further. Here I have created my own site and api too. Using this api, I was able to filter all the products from my site. Amazon, Flipkart and other ecommerce platforms do provide their api's to access their products. But there may be some criteria in order to get their api's If we get them, then we can create a platform where every product from all the sites can be accessed and we can make this filter the products from the entire ecommerce globe. Thus how we can save time and able to buy the best product.

REFERENCES

1.https://towardsdatascience.com/create-chatbot-using-rasa-part-1-67f68e89ddad
2.https://drive.google.com/file/d/1pJWOZS0LhLJuHh10O31fHS4RB2vdjiXx/view?usp=sharing
3.Ayedoun, E., Hayashi, Y., & Seta, K. (2015). A Conversational Agent to Encourage Willingness to Communicate in the Context of English as a Foreign Language. Procedia Computer Science, 60(1): 1433–14442
4.Egencia (2018). What is a Chatbot and How does it work? Retrieved March 9, 2019
5. Sproutsocial.com (2018). A complete Guide to Chatbots in 2018. Retrieved March 9, 2019
6. 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP).An automated system for educational domain
7. Chatbot Technologies and Challenges, 2018 First IEEE International Conference on Artificial Intelligence for Industries
8. Chatbot Magazine (2019). A Visual History of Chatbots. Retrieved March 9, 2019
9.Colace, F., De Santo, M., Lombardi, M., Pascale, L., Pietrosanto, A. Chatbot for E-Learning: A Cases Study. International Journal of Mechanical Engineering and Robotics Research Vol. 7, No. 5, September, 2018.
10.Egencia (2018). What is a Chatbot and How does it work? Retrieved March 9, 2019 .
11.Hattie, J. (2012). Visible learning for teachers: Maximizing impact on learning: Routledge. https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2
12.Lip ko, H. (2018). Meet Jill Watson: Georgia Tech's first AI teaching assistant. Retrieved on March 9, 2019.
13.Maruti Techlabs. Why can chatbots replace Mobile Apps immediately? Retrieved March 9, 2019
14.Nguyen, M. (2017). How artificial intelligence & machine learning produced robots we can talk to. Business Insider. Retrieved March 9, 2019
15.Sproutsocial.com . A complete Guide to Chatbots in 2018. Retrieved March 9, 2019
16,V Soft Consulting. (2019). 7 of the best Language-learning Chatbot Apps. Retrieved March 9, 2019