Report of Artificial Intelligence Chat Bot

regarding Meal Plan and Food Delivery

`

Course Name: **Artificial Intelligence Chat Bot**

Major : **Computer Science**

Author Name: **Hehui Gu**

Tutor : **Fan Zhang**

Date : **09/13/2019**

Summary

Artificial intelligence is now a very popular field, and artificial intelligence technology brings people more convenience and fun. Many large Internet companies are committed to developing smart chat bots using artificial intelligence technologies, such as Microsoft Xiaobing and Alibaba TmallGenie. This project is also to design an artificial intelligence chat bot. On the basis of daily conversation, it can meet the requirements of users to order food delivery and get the meal plan.

This project is deployed on Telegram. This project mainly realizes daily greetings, searching for restaurants, obtaining recommended foods, ordering food delivery, getting meal plan and searching for calories and various nutrient contents of food. People who are controlling their diet and who like to take a takeaway can use this chat bot to help themselves with this process.

This project uses python as the programming language. Use Rasa_nlu and spacy to train data to classify the intent of the message and extract the entity of the message. The main core technologies used are: match the pattern by Regular Expression, one round of multiple queries based on Incremental Slot Filling and Negation, multiple rounds of multiple queries based on State Machine and sending queries to API.

Keywords: Rasa_NLU, Spacy, Artificial Intelligence, Chat Bot, Natural Language Processing, State Machine, Negation, Incremental Filling Slot

Contents

I. Introduction

1.1 Background

Chat bots are currently a very hot artificial intelligence development and product direction. The original purpose of developing a chat bot was to pass the Turing test and allow the robot to communicate with people like a human .Many large Internet companies have invested in research and development related technologies, and have successively launched some chat bots, such as Apple Siri, Microsoft Cortana and Xiao Bing, and Alibaba TmallGenie. In the future, it is very likely that voice assistants can bypass the current many APPs to directly provide various services, such as querying weather, scheduled flights, ordering meals, device control of smart homes, voice control of in-vehicle devices, etc. Currently, most of them use independent APP forms to provide services. In the future, many applications that exist in the form of APP are likely to disappear, and they are hidden behind the chat bot.

A good chat bot should be highly robust and should be able to recognize multiple expressions of the same intent. Second, the response of the chat bot should be grammatically correct. Again, the chat bot's response should be fun, diverse, and not boring, and can express the same intent in many ways. A chat bot with specific functions should be able to accurately meet the needs of users.

Food Delivery is also a very popular service around the world. In the United States, there is Uber Eats, Yelp and GrubHub. In China, there is Meituan, Ele, and Dianping. As a daily service for many people, if we combine takeaways with smart chat robots, this technology will make life easier and more interesting for more people.These apps can recommend restaurants and foods based on the user's preferences and location, as well as precise information on the restaurant and food based on the user's screening

criteria, such as ingredients, calories and nutrient content of the food

1.2 My works

(1) Establish connections to a chatting API to interact with users

(2) Find the intent and extract entities of the requests sent by users

(3) Let the bot chat with users in some daily situations

(4) Let the bot have order food delivery and make meal plan regarding certain target calories functions.

(5) According to the request, send queries to related food delivery API to get the data, such like search restaurant, search food calories and food nutrition, and get some recommendations and image of food. Besides these searches may have some filter criteria.

(6) Get text, video, image response from food delivery API and do some processing.

(7) Send the the response to users through chatting API

II. Analysis of The Program

(1) Telegram has an open API and protocol free for everyone. Telegram delivers messages faster than any other application. Therefore I choose Telegram as my chat bot to interact with users.

(2) Rasa NLUis a very powerful tool for intent classification and entity extraction. So I use Rasa NLU and spacy to classify the intent and extract the entities of messages. In addition, I decide to use a little Regular Expression to make the results more accurate.

(3) When people have a little talk such like greeting ,saying goodbye or ask daily questions, we can easily recognize it by using Regular Expression to do keyword matching and choose our responses randomly from several answers.

(4) When people express the same intent but add more filter criteria for many times in one round, we use Incremental Filling Slot and Negation to store all paramas and negated params in this round. Therefore, we can easily get the answers under much filter criteria.

(5) In this program, there is no doubt that users will send many messages to the chat bot with different intents. As a result, we need to use state machine to make the bot covert its states when user's intent changes.

(6) I decided to use Recipe-Food-Nutrition API to get information of restaurants and food. Because this API is open and free. And it includes thousands of information of restaurants , food, food calories and food nutrition. Read the doc of this API, we can easily find ways to access, send query to this api and extract the useful information form the responses.

(7) We need to process the image, text and video different message types differently. When we receive image url from Recipe-Food-Nutrition API, we need to restore it and then send it by using send_photo() this method to send it to Telegram API. AS for text message, we use send_message() this method to send it to Telegram API.

III. Design and Implementation

3.1 Intent Recognition and Entity Extraction

I use spacy, rasa_nlu as the main method to classify the intent and extract entities of the sentences. Moreover, I use a little Regular Expression to help complete this process.

First of all, I construct the training data in json file,like this:

```
{
  "text": "I think 2000 calories one day is fine",
  "intent": "meal_plan",
  "entities": [
    {
      "start": 8,
      "end": 12,
      "value": "2000",
      "entity": "targetCalories"
    }
  ]
},
```

In this way, the intent of this sentence is “meal_plan”, and “2000” can be extracted as target calories to be used in the code. As for some sentences whose intent and entities are not easily identified by rasa_nlu, I use Regular Expression to extract and match the key words to find the intent and entities such like recognize ‘thankyou’ and ‘goodbye’ intention. Therefore, we can get more accurate intent and entity.

```
keywords = {
            'thankyou': ['thank','thanks','thx'],
            'goodbye': ['bye','see you','farewell']
           }
# Define a dictionary of patterns
regular_intent = {}

# Iterate over the keywords dictionary
for intent,keys in keywords.items():
    # Create regular expressions and compile them into pattern objects
    regular_intent[intent] = re.compile("|".join(keys))

def match_intent(message):
    matched_intent = None
    for intent, pattern in patterns.items():
        # Check if the pattern occurs in the message
        if re.search(pattern,message):
            matched_intent = intent
    return matched_intent
```

And I extract the name of food after ‘eat’ and ‘drink’ these two actions.

```
eats = ['drink', 'eat']
def entity_type(word):
    _type = None
    if word in eats:
        _type = "eat"
    return _type

def extract_food(doc):
    flag = False
    for word in doc:
        if flag == True:
            return word.text
        if entity_type(word.text) == "eat":
            flag = True
    return None
```

Finally, I train the data of the json file and call interpret() method to interpret the intent.

```
# Load the training data
training_data = load_data('train_calorie.json')

# Create an interpreter by training the model
interpreter = trainer.train(training_data)
```

3.2 One round of multiple queries based on incremental slot filling and negation.

According to some conversations that user will not repeat their intent in the next sentence but only add more restrictions to qualify the answer they will receive. Therefore, we need to have one or more params to document the all restrictions from the start of the conversations that have the same intent, which can be used by the following responses.

I use params and negated params to record the normal restrictions and negated restrictions to filter the searching results. Params can be extracted by the trainer and then the negated parmas will be identified and added into negated parmas. When the intent is ‘meal plan’, the user can add restrictions to find suitable meal plans.

```
if intent == "meal_plan":
    # Extract the entities
    entities = interpreter.parse(message)["entities"]
    ent_vals = [e["value"] for e in entities]
    negated = negated_ents(message, ent_vals)
    for ent in entities:
        if ent["value"] in negated and not negated[ent["value"]]:
            neg_params[ent["entity"]] = str(ent["value"])
        else:
            params[ent["entity"]] = str(ent["value"])
    # Find the meal plan
    meal_plans,params,neg_params,rate = find_meal_plans(params, neg_params)
    n = min(len(meal_plans), 3)
    # Return the appropriate response
    return responses[n].format(*meal_plans),response2,new_state, pending,params,flag,neg_params,rate
```

Identify the negated entities:

```
if "not" in chunk or "n't" in chunk:
    result[ent] = False
```

3.3 Multiple rounds of multiple queries based on state machine

In order to build a 'smarter' bot, we need to let the bot can answer multiple rounds of queries with different intends. Consequently, we can use state machine to transform the state of the bot to another state, when the intent of the query changes.

In addition, we have a parma called pending_state .When the user asks for ordering food, the pending_state will be assigned AUTHED to ask users' phone number to log in or the user can not continue to order food delivery.

```
# Define the policy rules
policy_rules = {
    #(INIT, "ask_condition"): (INIT, "I am a bot to help you make a meal plan or order some food", None),
    (INIT, "order"): (INIT, "Please offer your phone number at first", AUTHED),
    (INIT, "ask_condition"): (INIT, "I am a bot to help you make a meal plan or order some food", None),
    (INIT, "plan"): (PLAN_CAL, "How many calories are you planning to consume in one day ?", None),
    (PLAN_CAL, "meal_plan"): (MEAL_PLAN, None, None),
    (MEAL_PLAN, "meal_plan"): (MEAL_PLAN,None, None),
    (MEAL_PLAN, "pie"): (INIT, None, None),
    (INIT, "number"): (AUTHED, "You can order food now",None),
    (AUTHED, "order"): (CHOOSE_RESTAURANT, "Which restaurant would you like to order from?", None),
    (CHOOSE_RESTAURANT, "restaurant"): (ORDER_FOOD, "What food would you like in {0} ?", None),
    (ORDER_FOOD, "recommand"): (ORDER_FOOD, "How about {0} ?", None),
    (DELIVERY, "ask_calories"): (INIT, "The calories are {0}", None),
    (ORDER_FOOD, "food"): (DELIVERY, "OK, you can get your delivery a few minutes later", None),
    (INIT, "greet"): (CHAT, None,None),
    (CHAT, "chat"): (CHAT, None,None),
    (CHAT, "ask_condition"): (INIT, "I am a bot to help you make a meal plan or order some food",None)
}
```

In respond method, I call policy rules in this way.

```
new_state, response, pending_state = policy_rules[(state, interpret(message))]
```

3.4 Send Query to Recipe-Food-Nutrition API

In this program, the bot need to search the food items of pointed restaurant, search the calories of pointed food and search the suitable meal plans regarding target calories and other restrictions.

I pass the extracted params to this method as a params of the query which will be sent to Recipe-Food-Nutrition API. Then I parse the response to json object and extract the useful information from the whole response. (the name of food and the image of the food to give more direct impressions)

```
def find_food(params):
    url = "https://spoonacular-recipe-food-nutrition-v1.p.rapidapi.com/food/menuItems/search"
    querystring = {"offset":"0","number":"1","minCalories":"0","maxCalories":"5000","minProtein":"0",
                   "maxProtein":"100","minFat":"0","maxFat":"100","minCarbs":"0","maxCarbs":"100","query":""}
    querystring['query'] = params
    headers = {
    'x-rapidapi-host': "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com",
    'x-rapidapi-key': "1530809090mshbd1bfa86ad13155p19a75djsn12b4d43f7fcb"
    }
    response = requests.request("GET", url, headers=headers, params=querystring)
    response = response.json()
    return response['menuItems'][0]['title'],response['menuItems'][0]['image']
```

3.5 Make pie plot of the nutrition content of the meal plan

Using params to record the content of nutrition, so in next state, if user asks for the pie plot of nutrition content, the bot can easily show it. And we save this plot as one picture so the bot can easily send it to the user.

```
for ra in rate:
    if ra != 'calories':
        labels.append(ra)
        sizes.append(rate[ra])
    else:
        response = rate[ra]
colors = ['pink','yellowgreen','skyblue']
plt.pie(sizes,labels=labels,colors=colors,autopct='%2.0f%%')
plt.axis('equal')
plt.legend()
plt.show()
picture.savefig("pie.png")
return response
```

3.6 Connect with Telegram API

First of all, bind the bot with my own telegram API key:

```
bot = telebot.TeleBot('[redacted]')
```

Then, bind send_reply() with bot.message.handler(). Therefore, when there is a message sent by users, send_reply() will be triggered and start to process the received message.

```
@bot.message_handler()
def send_reply(message2):
    # Get the bot's response to the message
```

We use send_photo() to send reply of the image to the user, and use send_message() to send text reply to the user.

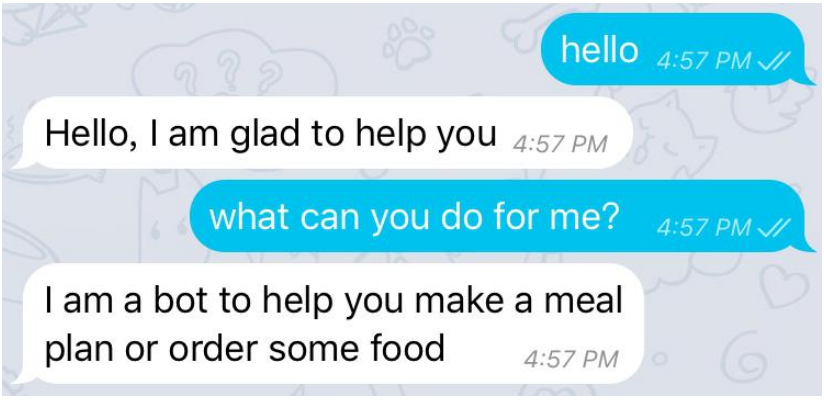
```
    bot.send_photo(message2.chat.id, photo=open('pic.jpg', "rb"))
else:
    bot.send_message(message2.chat.id, response2)
```

Finally, we open the connection to Telegram API

```
bot.polling()
```

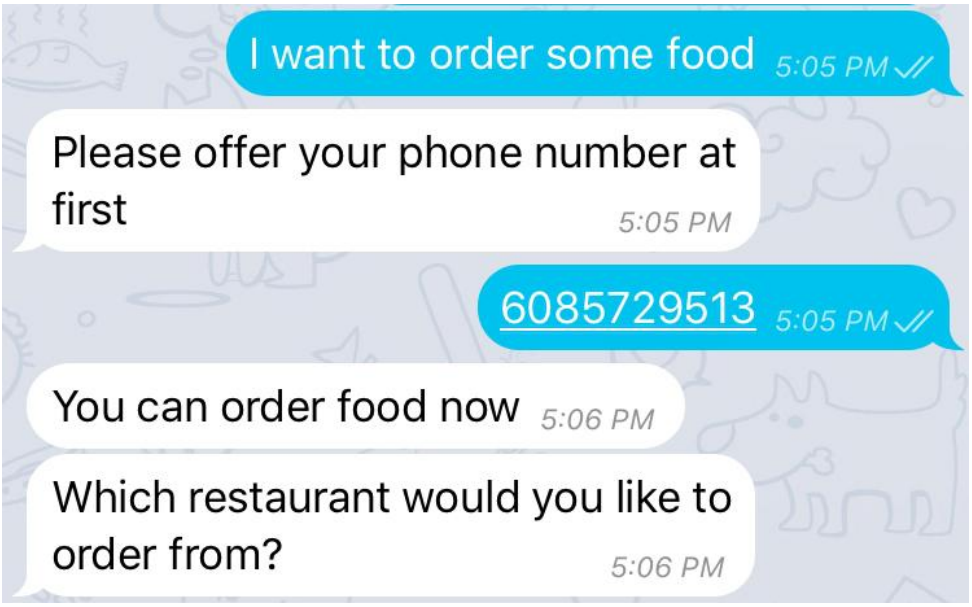
IV. Results

4.1 Chat with Fun

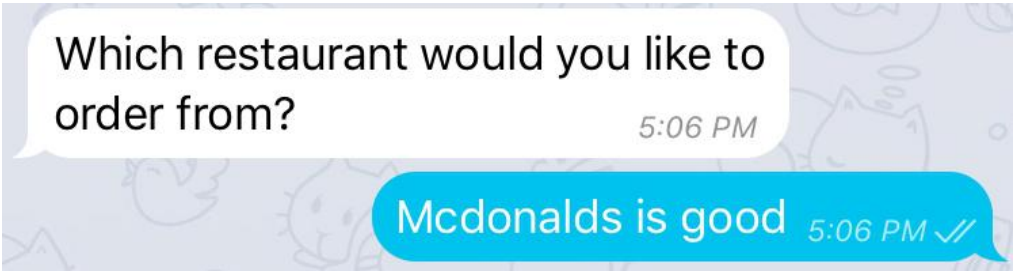


4.2 Multiple Random Answers

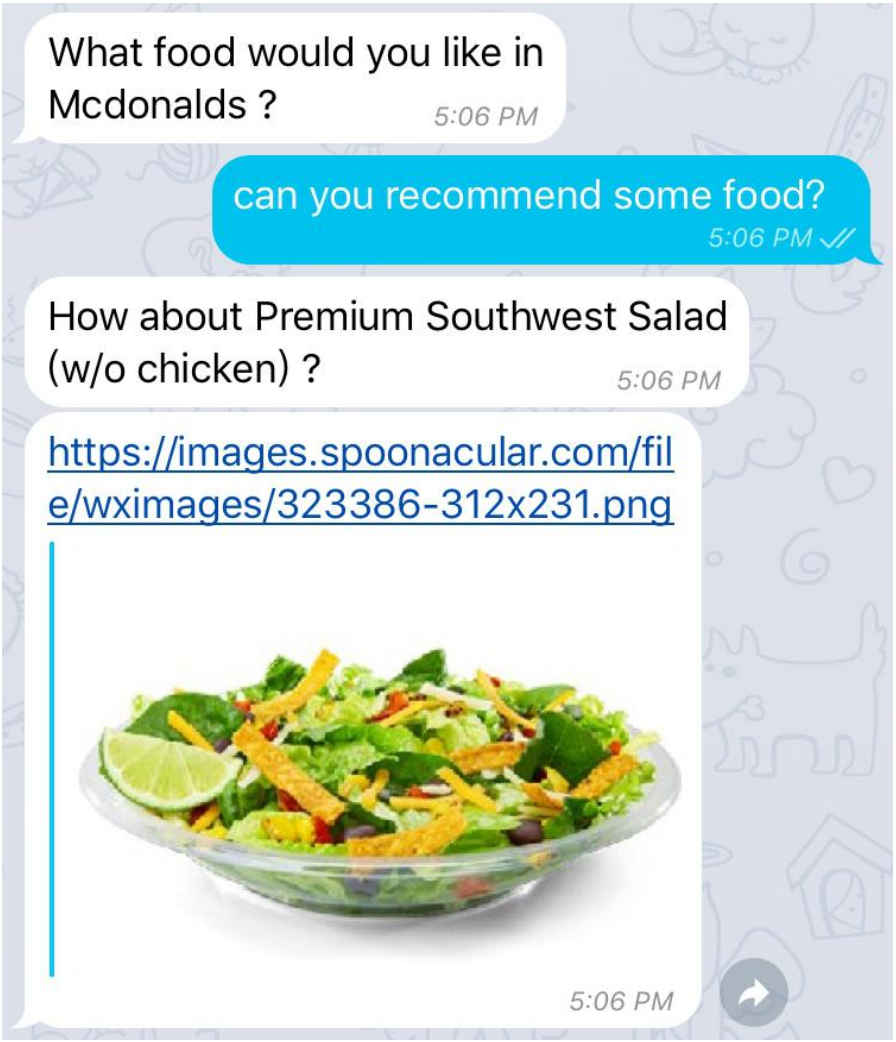
4.3 Need to log in using phone number before ordering food



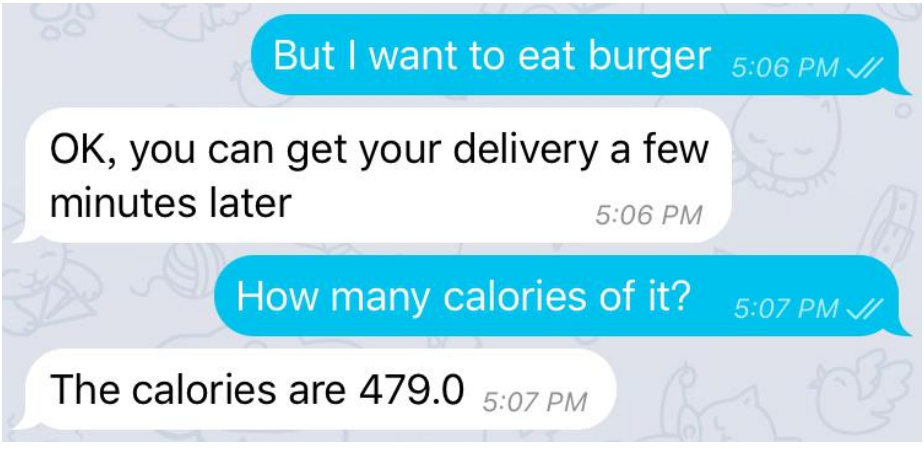
4.4 Choose the restaurant of food delivery after login



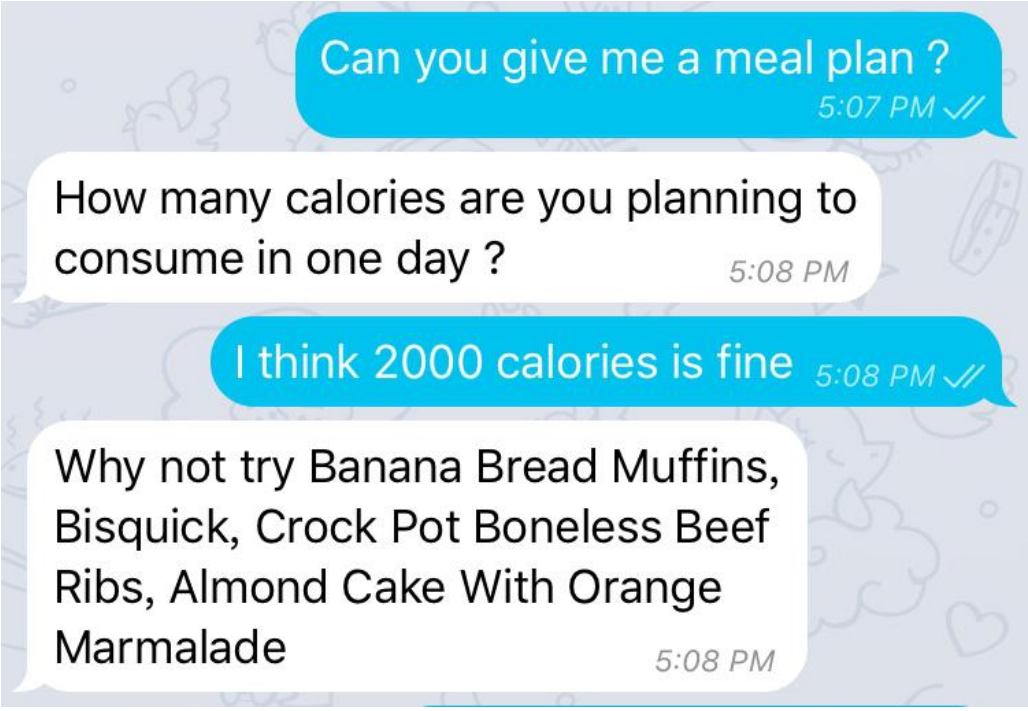
4.5 Ask for some recommendations of food in this restaurant



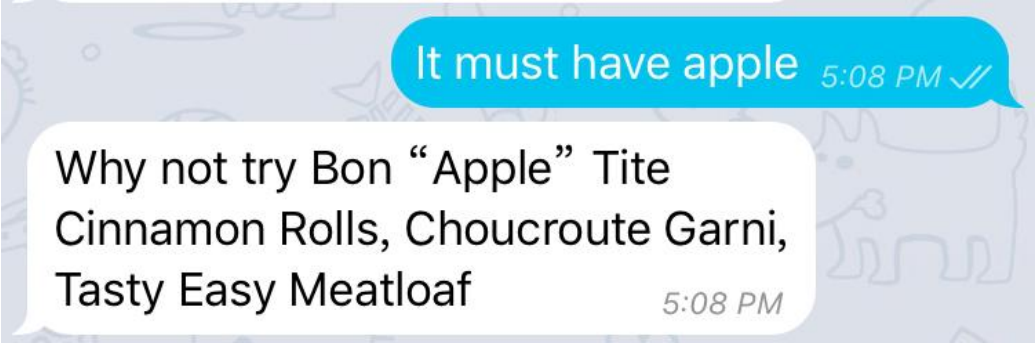
4.6 Order other food and ask about the calories



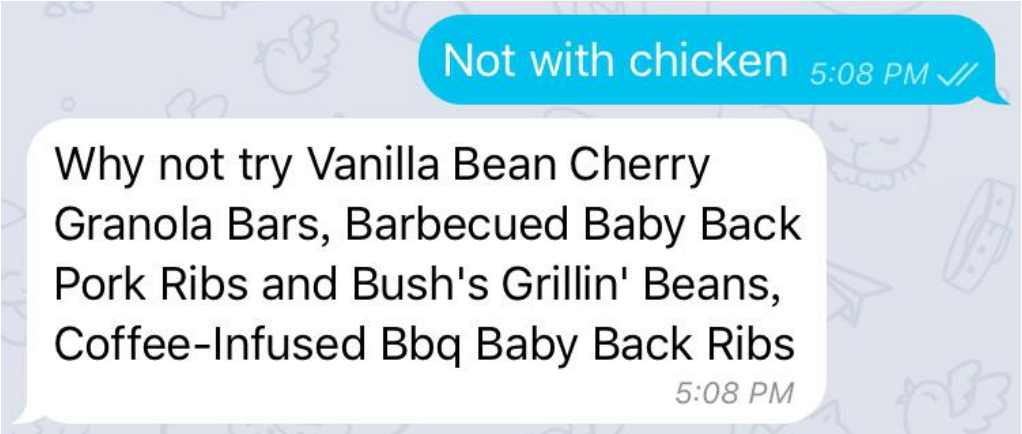
4.7 Asking for making a meal plan regrading certain calories



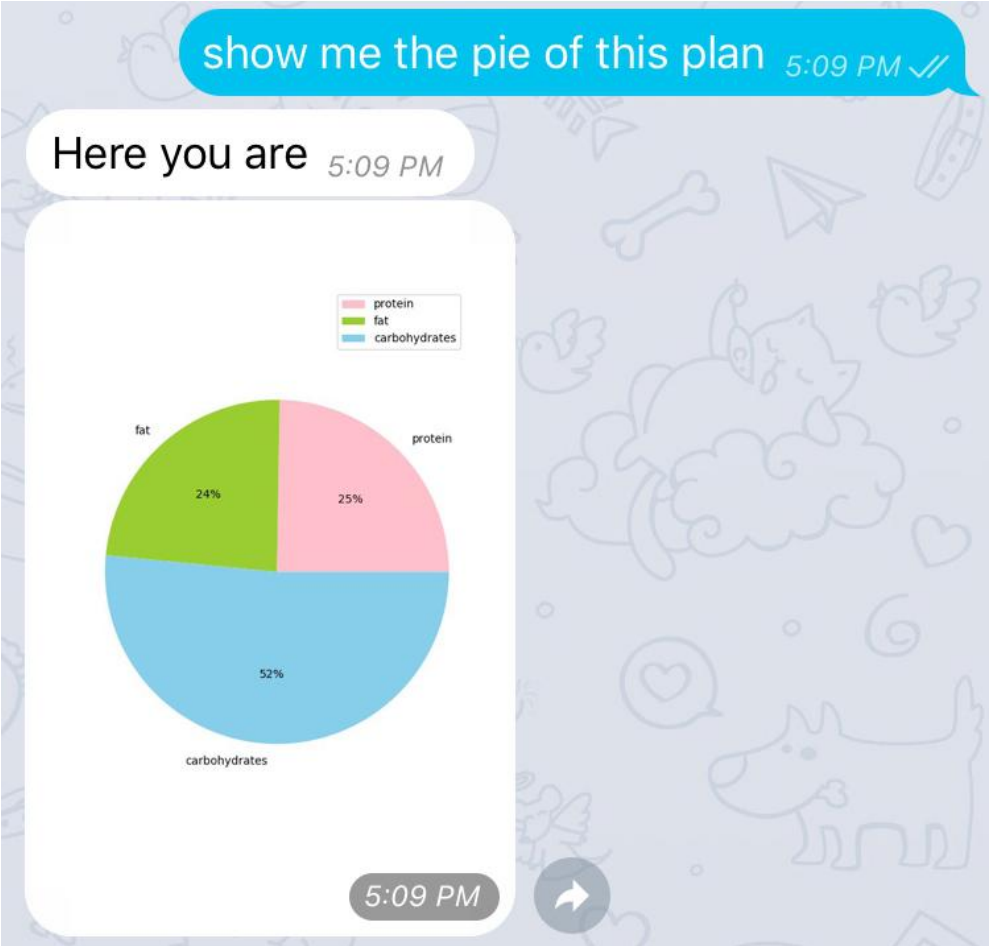
4.8 Using incremental slot filling



4.9 Identify the negation and filtering with excluded slots



4.10 Show user the pie plot of the nutrition of this meal plan



V. Conclusion

Through this remote lecture and completion of the artificial intelligence chat robot project, my understanding of the field of artificial intelligence has been further improved, and my Python code capabilities have been greatly improved.

In the teacher Zhang's class, I systematically learned a lot of new knowledge in the field of artificial intelligence, such as pattern matching with regular expression, keyword extraction, classification of user intent with nearest neighbor method and support vector machine, and analysis of user intention with Rasa NLU, extracting entities, database query with python, single-round multiple incremental queries based on incremental filling slot and negation, and multi-round multiple queries based on state machine. When Teacher Zhang lectured, he first talked about theoretical knowledge, and then took us to study the corresponding code, so that we have a very deep understanding of the corresponding technology. And in each assignment, we practiced the lessons taught in the class which laid the foundation for a complete chat robot project.

Teacher Zhang not only taught us the knowledge, but also helped us to open our horizons and broaden our thinking which can guide our future studies in the fields of computer, artificial intelligence and natural language processing. The teacher introduced us about the development status of artificial intelligence, the maturity and application of various technologies, and recommended many related papers and conferences to us as materials for further study in the future.

In terms of Python's capabilities, I learned to call a lot of methods and packages, and learned to access the API. The fluency of programming skill has increased. I solved the problem when I finished my homework and completed the project. I asked the teacher for help or found information from the Internet. Through this process, my confidence in writing code and finnishing projects has increased. In the project, I used the knowledge I learned in class which consolidated my understanding of this knowledge. In the end, after I deployed the code to the Telegram, I gained a great sense of accomplishment and be inspired me to continue to learn natural language processing and other artificial intelligence knowledge.