# Natural Language Processing – Assignment 3

**Name : Pavithra Ramasubramanian Ramachandran**
**Student ID : R00183771**

**Instructions to run the code:**

I have made edits to the code provided in Canvas to build my bot.

There are four code files :

- chatbot.py
- data.py
- config.py
- model.py

Given movie data corpus : Retrieved from Canvas
Additional corpus file : conversations_data.txt

Execution Instructions:
In command line terminal :

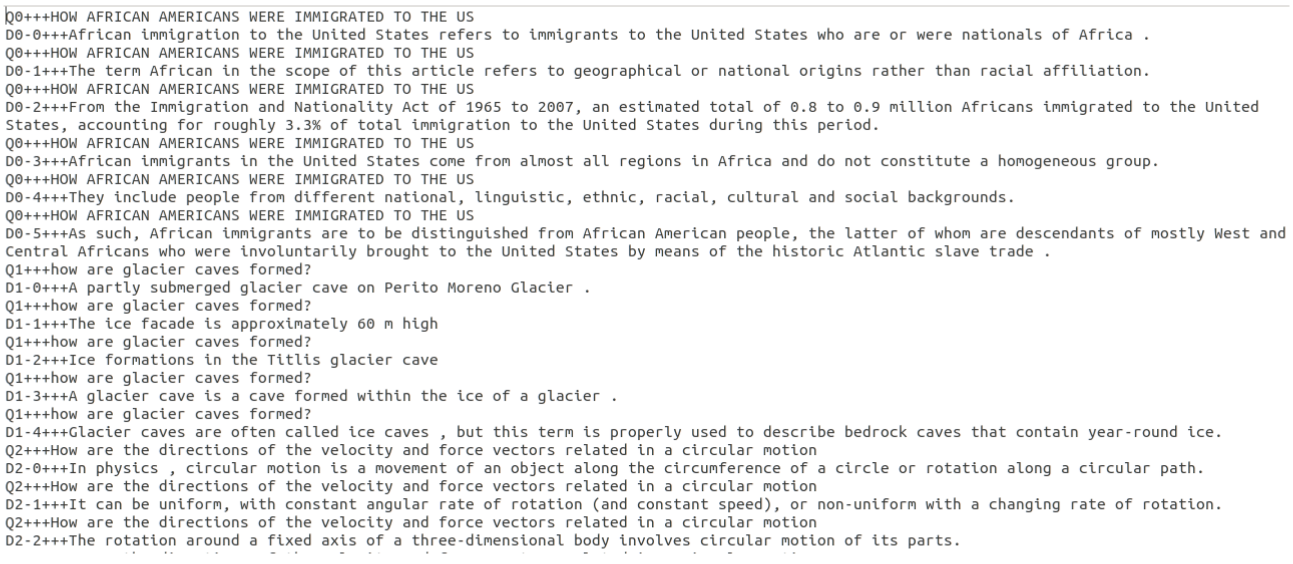
```
> python data.py
> python chatbot.py –mode chat
```

## Improvements done on chatbot code:

### 1. Train on multiple datasets (Mandatory)

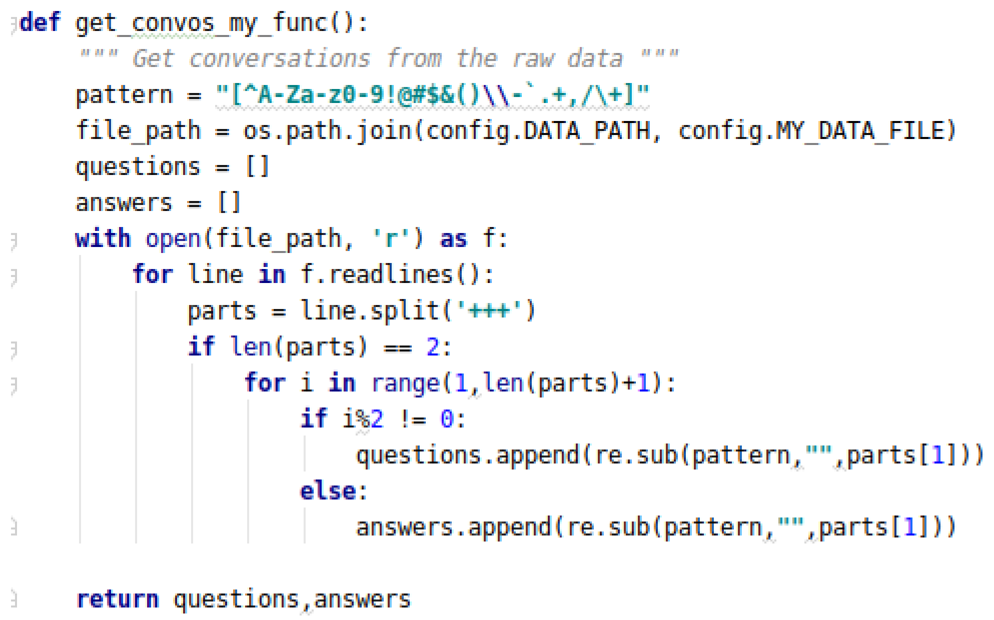
Dataset source : https://www.microsoft.com/en-us/download/details.aspx?id=52419

This data consists of general questions asked on Wiki and a couple of answers relevant to the questions.This dataset was trained alongside the movie data. The downloaded file was in the .tsv format. A little bit of processing was done to convert the file to .txt with the following format as shown in the below screenshot so that pre-processing and training the data would become easier.



```
Q0+++HOW AFRICAN AMERICANS WERE IMMIGRATED TO THE US
D0-0+++African immigration to the United States refers to immigrants to the United States who are or were nationals of Africa .
Q0+++HOW AFRICAN AMERICANS WERE IMMIGRATED TO THE US
D0-1+++The term African in the scope of this article refers to geographical or national origins rather than racial affiliation.
Q0+++HOW AFRICAN AMERICANS WERE IMMIGRATED TO THE US
D0-2+++From the Immigration and Nationality Act of 1965 to 2007, an estimated total of 0.8 to 0.9 million Africans immigrated to the United
States, accounting for roughly 3.3% of total immigration to the United States during this period.
Q0+++HOW AFRICAN AMERICANS WERE IMMIGRATED TO THE US
D0-3+++African immigrants in the United States come from almost all regions in Africa and do not constitute a homogeneous group.
Q0+++HOW AFRICAN AMERICANS WERE IMMIGRATED TO THE US
D0-4+++They include people from different national, linguistic, ethnic, racial, cultural and social backgrounds.
Q0+++HOW AFRICAN AMERICANS WERE IMMIGRATED TO THE US
D0-5+++As such, African immigrants are to be distinguished from African American people, the latter of whom are descendants of mostly West and
Central Africans who were involuntarily brought to the United States by means of the historic Atlantic slave trade .
Q1+++how are glacier caves formed?
D1-0+++A partly submerged glacier cave on Perito Moreno Glacier .
Q1+++how are glacier caves formed?
D1-1+++The ice facade is approximately 60 m high
Q1+++how are glacier caves formed?
D1-2+++Ice formations in the Titlis glacier cave
Q1+++how are glacier caves formed?
D1-3+++A glacier cave is a cave formed within the ice of a glacier .
Q1+++how are glacier caves formed?
D1-4+++Glacier caves are often called ice caves , but this term is properly used to describe bedrock caves that contain year-round ice.
Q2+++How are the directions of the velocity and force vectors related in a circular motion
D2-0+++In physics , circular motion is a movement of an object along the circumference of a circle or rotation along a circular path.
Q2+++How are the directions of the velocity and force vectors related in a circular motion
D2-1+++It can be uniform, with constant angular rate of rotation (and constant speed), or non-uniform with a changing rate of rotation.
Q2+++How are the directions of the velocity and force vectors related in a circular motion
D2-2+++The rotation around a fixed axis of a three-dimensional body involves circular motion of its parts.
```

Pre-processing of the new corpus was done in the data.py file. Below is the code snippet for the same:



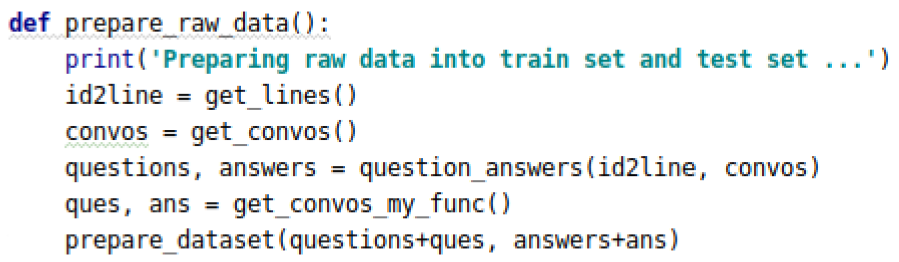
```
def get_convos_my_func():
    """ Get conversations from the raw data """
    pattern = "[^A-Za-z0-9!@#$&()\\-`.+,/\+]"
    file_path = os.path.join(config.DATA_PATH, config.MY_DATA_FILE)
    questions = []
    answers = []
    with open(file_path, 'r') as f:
        for line in f.readlines():
            parts = line.split('+++')
            if len(parts) == 2:
                for i in range(1,len(parts)+1):
                    if i%2 != 0:
                        questions.append(re.sub(pattern,"",parts[1]))
                    else:
                        answers.append(re.sub(pattern,"",parts[1]))

    return questions,answers
```

An additional function get_convos_my_func() was defined to pre-process the data in the new corpus. The following ore-processing steps were implemented in the code:

- Since the data has a lot of junk characters, characters that belong to other languages, a regex pattern ws used to remove these junk data from the data.
- Lines read from the .txt file are split by '+++' to obtain questions and answers in alternate line of the file. They are stored in two lists - 'questions' and 'answers' respectively.

After pre-processing of new data, it is appened to the raw data as this raw data will be sent to other functions to train and test. The below code snippet includes the get_convos_my_func() function and appends the questions and answers to the arguments sent to prepare_dataset function.



```
def prepare_raw_data():
    print('Preparing raw data into train set and test set ...')
    id2line = get_lines()
    convos = get_convos()
    questions, answers = question_answers(id2line, convos)
    ques, ans = get_convos_my_func()
    prepare_dataset(questions+ques, answers+ans)
```

## 2. Make your chatbot remember information from the previous conversation

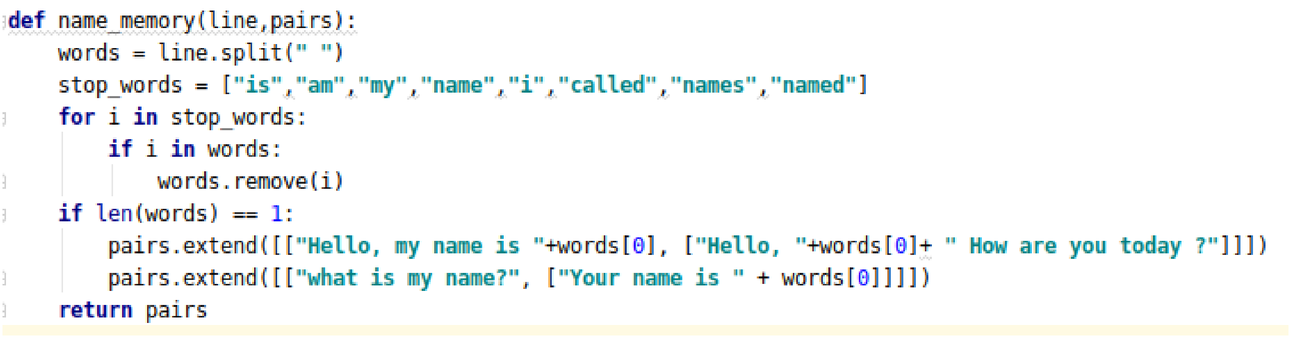
My bot can remember chat details from previous conversations, in specific, my bot can remember the name of the user if the user specifies his/her name. This functionality was implemented using a rule based approach where, if the user specifies a sentence which consists of the word 'name', the the bot will address the user with his/her name in response.

**Limitations:**

- The user should mandatorily mention 'name'in the sentence while mentioning name.
- Any other sentence that contains the word 'name' will be misinterpreted to be conveying th ename of the user.

- There should be no junk words or spelling errors

Below is the code snippet of this functionality implemented in chatbot.py file:



```python
def name_memory(line,pairs):
    words = line.split(" ")
    stop_words = ["is","am","my","name","i","called","names","named"]
    for i in stop_words:
        if i in words:
            words.remove(i)
    if len(words) == 1:
        pairs.extend([["Hello, my name is "+words[0], ["Hello, "+words[0]+ " How are you today ?"]]])
        pairs.extend([["what is my name?", ["Your name is " + words[0]]]])
    return pairs
```

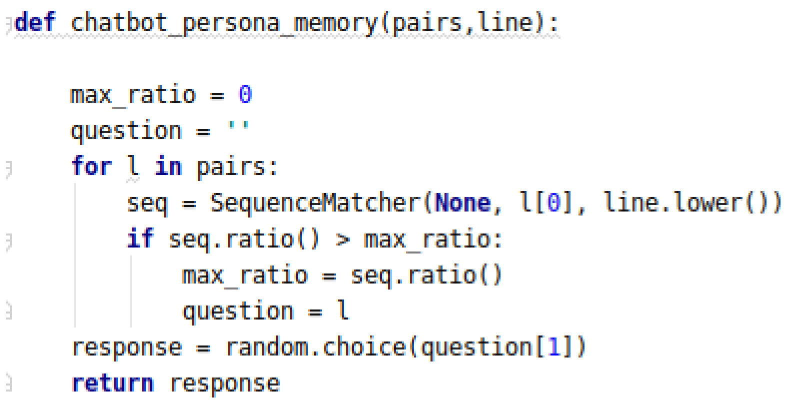
In the above function, stop words such as ‘name’, ‘is’ , ‘am’ are removed from the line and the remaining word is considered as the name and rules with the name of the user are appended to the rule list.

### 3. Create a chatbot with personality

I have created a bot by
- Name : Jack
- Age : 1 week
- Hometown : Cork
- Created by : Pavithra
- Job : To help

Used rule based approach to implement this. To do so, I created a list of lists to store all the rules and also used the SequenceMatch package. The SequenceMatch package checks ratio match between the input line by user and the rule-based data structure’s rules. If the match is high, then it implements rule-based bot, otherwise it implements corpus-based chatbot. The rule with the max_match ratio is chosen and a response is given to the user based on this rule.

The chatbot_persona_memory() function is used to define this functionality in chatbot.py file. The code snippet is as below.



```python
def chatbot_persona_memory(pairs,line):

    max_ratio = 0
    question = ''
    for l in pairs:
        seq = SequenceMatcher(None, l[0], line.lower())
        if seq.ratio() > max_ratio:
            max_ratio = seq.ratio()
            question = l
    response = random.choice(question[1])
    return response
```

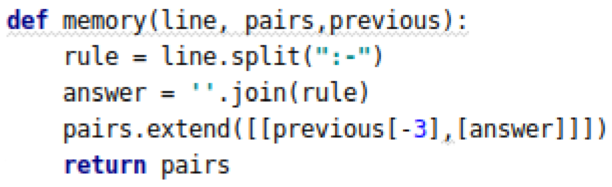
### 4. Create a feedback loop that allows users to train your chatbot

The chatbot responds to the user’s queries by either retrieving data from the corpus or from the rules defined based on the SequenceMatch ratio. However, in certain instances, these responses may be wrong. In such situations, the user should respond by saying ‘ wrong answer/incorrect answer’ and provide the correct answer to the bot. The bot will remember this answer and respond correctly in future iterrogations.

**Flow of code:**

User: ‘Asks question’
Bot: ‘Provides wrong answer’
User: ‘Wrong answer’
Bot: ‘Please provide the correct answer preceeded by ‘:-’ ‘
User : ‘:-provides the correct answer’
Bot: Thanks for that!
User: ‘Asks Same question’
Bot:’Responds with the correct answer’

This functionality is implemented using the following code snippet:



```
def memory(line, pairs,previous):
    rule = line.split(":-")
    answer = ''.join(rule)
    pairs.extend([[previous[-3],[answer]]])
    return pairs
```

As shown in the above snippet, the correct answer that is obtained from the user is appended to the list of rules defined. Therefore, in future if the user asks the same question, the chatbot will be able to answer appropriately.

**Limitations:**
The conversation should happen as defined in the ‘Flow of code’ section mandatorily.

## <u>**Sample Output:**</u>

```
HUMAN ++++ hello
BOT ++++ Hey there
HUMAN ++++ my name is pavithra
BOT ++++ Hello, pavithra How are you today ?
HUMAN ++++ what is your name?
BOT ++++ My name is Jack and I'm a chatbot .
HUMAN ++++ how old are you?
BOT ++++ I am 1 week old
HUMAN ++++ who is the president of usa?
BOT ++++ <unk>
HUMAN ++++ wrong answer
BOT ++++ Apologies for the error! Please give the correct response preceeded by ':-'
HUMAN ++++ :-Donald Trump
BOT ++++ <unk> - <unk>
HUMAN ++++ Who is the president of usa?
BOT ++++ Donald Trump
```

HUMAN ++++ quit
BOT ++++ Bye for now. See you soon :)