Coursework1 Report

Name: Gao Sihang ID:alysg10

Introduction:

My system is basically divided into five parts: ‘chat’, ‘question and answer’, ‘query date’, ‘query weather’ and ‘playing games’. They corresponds to different modules by different functions. Users can go back to the previous level menu to select other features after entering a feature. Users can also input ‘bye’ at any time to end the conversation.

I chose the ‘question and answer’ feature because moodle provide a table of relevant data for question and answer, which saved me a lot of time to build a corpus. Weather and date queries were something that everyone would care about everyday. Playing games can make people feel relaxed after their work. So, as a daily conversational robot, I think these functions are necessary. Finally, the most basic of chatbots is the chat function, so I think this is also essential.

Background:

I read the "A Tool of Conversation: Chatbot" paper, which describes the purpose and functionality of chatbots and some basic principles. However, he tends to use java programming, and this article ignores users input mistakes which lead to the robot can not normally extract labels from the corpus this situation. I think setting up an error correction function is conducive to make chatbots more efficient and more convenient for users to use. I also looked at the contents of ‘lab1’, ‘lab2’, ‘lab3’ and learned some ways to preprocess text. However, comparing to ‘Sklearn’ machine learning algorithm, using ‘Tensorflow.Keras’ deep learning algorithms are more convenient and more efficient to train with gpu. Using gpu to train model can save a lot of time.

Functionality:

Chatbots can first identify user information. Users can select specific modules to implement ‘chat’, ‘question and answer’, ‘Query time’, ‘query weather’ and ‘playing games’. In addition, spelling errors entered by the user can be effectively corrected during the conversation. Users can input some information to enter each function, return to the previous step or end the chat. If the input content can be answered in other functions, a paragraph will be output to remind the user to communicate with other modules. Checking the weather can check the current weather in real time and the weather of the next day. At the same time, the city can also choose any location in the UK. Even if the user wants to travel to a certain city the next day, the robot can accurately give the weather conditions. You can play the game of dice toss. It can relax you when you are tired at work.

Originality:

I converted all the corpora into json format, and the corpus I made is also in json format, which is convenient to call and view. There will be no repeated output of the question content when the web version asks a question. I added an error correction function to make the chatbot more efficient and convenient. I use the keras deep learning method to refine the data with the machine learning model itself, thereby training the model, saving the time of artificial feature extraction, and it is more suitable for the extraction of a large number of data features.

Implementation:

First, I need to configure the environment, download the library and various functions. Especially when downloading tensorflow.Keras, you need to download the gpu version to improve the training speed, but it is relatively troublesome to install. The cpu version of tensorflow.Keras can also be used to

train the model. However, if you encounter a relatively large corpus, the efficiency may be very low. The training time is very long, which is not conducive to debugging the network to find the best network model.

I used the corpus on moodle to build a json format corpus, so that each question and answer has a corresponding label. Then I built the chat json corpus. I wrote the conversations that I often use in the online and daily conversations into the json file and tagged them.

I perform different preprocessing on these two corpora, delete the corresponding stopwords. Then I merge the data of the two corpora, prepare the bag-of-words, and import the tensorflow.Keras training corpus. The number of nodes of the neural network is determined by the number of training samples, and finally the training results are saved in an h5 file for later use.

```
151/151 [==============================] - 1s 6ms/step - loss: 0.0078 - accuracy: 1.0000
Epoch 150/150
151/151 [==============================] - 1s 6ms/step - loss: 0.0079 - accuracy: 1.0000
Done

Process finished with exit code 0
```

In **error correction function**, I put the questions in the chat into a notebook as a corpus.

Convert all data to lowercase and remove special characters and read the existing database and count. Calculate the probability of words appearing in the database.

Edit the word. If the word is in the corpus, output the corresponding word directly. If the word is not in the corpus, there may be a problem with the spelling of the word. Some modifications are required, such as delete a letter, swap front and back letter, substitutes a new letter or insert a letter. Put all the possible original words together in a random order, first judge whether these original words are in the corpus, extract the words in the corpus and then make a probability judgment, and select the word with the highest probability of appearing in the

corpus as the output. If these original words are not in the corpus, output the original wrong form.

In the main function, first, I import and read the previous json file, pk file and the previous training model. Initialize the chat bot, if the input is "bye", output "bye, good luck!" and then end the dialogue. You can also enter ‘chat’, ‘question and answer’, ’date’, etc. to select the corresponding function to start the relevant dialogue. The first function is chat. When this function is selected, the robot will first output ‘ok, let’s have a chat’ and then start chatting. The user enters a sentence, first preprocess and correct the sentence, and then prepare the Bag-of-Words and make a match. I use the previously trained model to calculate the label probability corresponding to this sentence. If the probability is greater than 0.55, then output a certain answer corresponding to the label with the highest probability. Since I put two separate data sets together for training, the robot can search all corpora to find the answer. **If it is found that the answer with the highest probability has a corresponding answer under 'question and answer' function**, then answer "Please 'back to previous step' and choose 'question and answer' to ask question". If the corresponding probability label is not found, answer "I am sorry! I don't understand you" and enter "bye" to end the conversation directly. The code of ‘question and answer’ is similar to that of ‘chat’, and the function is also similar, except that the corpus used is different.

The ‘time’ function can directly query the real-time date by calling the time function, and automatically return to the previous menu after the query.

The weather function can directly crawl the webpage in real time through the request function and the beautifulsoup library. I first find the url address, and then I can change the url address by entering different regions and different times. In the process of crawling the webpage, I first obtain the webpage information and analyze it. Then, I found the location of the desired temperature and weather in the html file. Finally, I use python to extract the string at that position for output.

The ‘play games’ function is to play the dice game. First, user can enter the amount of the account. Then user can determine the guess and enter the bet for this time. The robot randomly outputs a number from 1-6, if it is ‘1-3’, it is 'small', if it is ‘4-6’, it is 'big'. If the account is none, the game is over. User can also enter ‘stop’ to end the game.

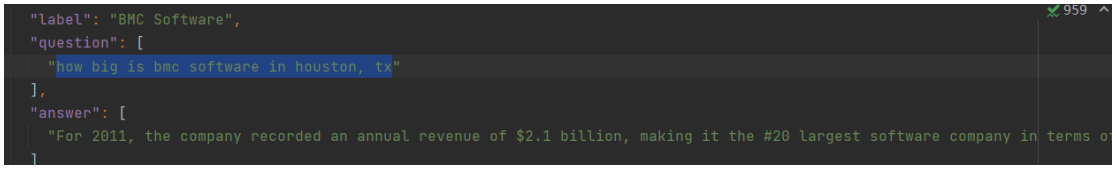
Evaluation:

I asked my friend to perform a test. First, he entered his name and got a response related to his name.

```
The bot is on
BOT: Hi, what is your name?
my name is Wang Zhifei
BOT: Hi,  Wang Zhifei
BOT:please input 'chat','question and answer','date' 'play games' and 'weather'
chat
BOT: Ok,  Wang Zhifei,Let's have a chat
```

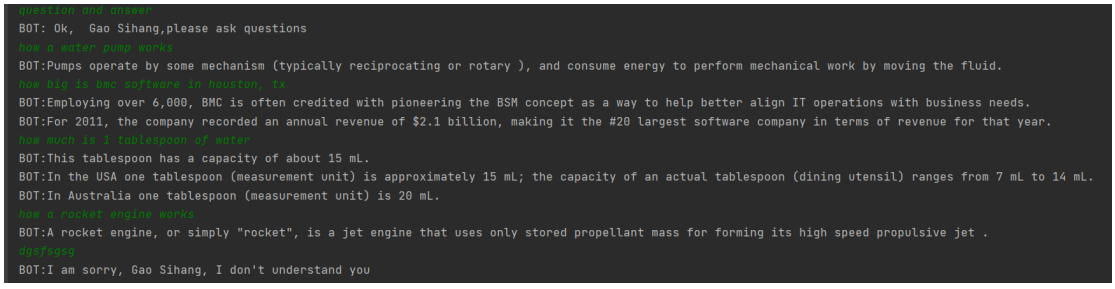
Then he chose the 'chat' function to test. For the questions in the corpus, the robot responded very well, and the dialogue outside the corpus would reply ‘I am sorry! I don't understand you’. If there is a question and the answer of the question is in ‘question and answer’, robat will reply "Please back to previous step and choose 'question and answer' to ask question". If you want to end the 'chat' and perform the 'question and answer' function, enter 'back to previous step' to return to the menu at the beginning.

```
BOT:please input 'chat','question and answer','date' 'play games' and 'weather'
chat
BOT: Ok,  Wang Zhifei,Let's have a chat
how are you
BOT:I am fine
do you play football
BOT:I don't know how to play
do you like wine
BOT:if i could drink i probably would
vdsgsg
BOT:I am sorry,  Wang Zhifei, I don't understand you
hsgfsgsdhgd
BOT:I am sorry,  Wang Zhifei, I don't understand you
what should we wear in winter
BOT:I am sorry,  Wang Zhifei, I don't understand you
how big is bmc software in houston, tx
BOT:Please 'back to previous step' and choose 'question and answer' to ask question
thank you
BOT:you are welcome
back to previous step
BOT: please input 'chat','question and answer','date' 'play games' and 'weather'
```



(It is not in ‘chat’(output2. json) but in ‘question and answer’(output. json))

Then he tested the ‘question and answer’ function. The result was basically the same as ‘chat’, and a good answer could be obtained.



Next, he tested the time function and got the specific time, year, month, and day.

```
back to previous step
BOT: please input 'chat','question and answer','date' 'play games' and 'weather'
date
Thu Nov 18 00:44:23 2021
BOT:please input 'chat','question and answer','date' 'play games' and 'weather'
```

He continued to test the weather function. He tested two groups, namely the weather in Nottingham today and the weather in London tomorrow, and both got accurate responses.

```
BOT:please input 'chat','question and answer','date' 'play games' and 'weather'
weather
BOT:Which city do you want to search?
London
BOT:Which day do you want to search,today or tomorrow?
tomorrow
  +12°C     Overcast, no precipitation
BOT:please input 'chat','question and answer','date' 'play games' and 'weather'
weather
BOT:Which city do you want to search?
I want to search Nottingham
BOT:Which day do you want to search,today or tomorrow?
today
  +6Â°C     Overcast, no precipitation
BOT:please input 'chat','question and answer','date' 'play games' and 'weather'
```

Finally, the game function was tested. The operation was normal, and the user had fun.

```
BOT:please input 'chat','question and answer','date' 'play games' and 'weather'
play games
BOT:Welcome to play "Toss the dice"
BOT:Please enter the initial amount: 6
BOT:you have 6 pound
BOT:big or small or stop:3
BOT:your input is wrong, please input again
BOT:you have 6 pound
BOT:big or small or stop:big
BOT:bet:3
5
BOT:you win!
BOT:you have 9 pound
BOT:big or small or stop:big
BOT:bet:9
5
BOT:you win!
BOT:you have 18 pound
BOT:big or small or stop:big
BOT:bet:18
5
BOT:you win!
BOT:you have 36 pound
BOT:big or small or stop:big
BOT:bet:36
2
BOT:you lose!
BOT:Game over
BOT:please input 'chat','question and answer','date' 'play games' and 'weather'
```

```
bye
BOT: Bye!  Wang Zhifei, good luck!

Process finished with exit code 0
```

After comprehensive evaluation, the fluency and accuracy of the robot were relatively high, and the user was satisfied. However, because the corpus is not large enough, many questions outside the corpus cannot be answered very well. In this regard, the users are not very satisfied, but I believed that this problem can be solved by filling the corpusin the future.

In addition, the user said that in addition to time and weather, there were not many other functions used in daily life, such as buying tickets, ordering takeaways, and other functions. This is mainly because I have no experience in crawling web pages, and I have no experience in crawling web pages. The learning time is relatively short, and it is impossible to extract accurate information from more complex web pages in a short time. I think this is not a problem of nlp itself, but a simple problem of web page information extraction. In the future, through more in-depth study of web page information extraction, this problem will be solved.

Finally, in general, the user was satisfied with the system as a whole.

Discussion:

The results of the system test are basically the same as expected. It can perform chat, question and answer, query dates, query weather and playing games. In real applications, I think it can relax people, relieve anxiety and solve users based on the corpus. You can also inquire about the time and weather, which is convenient for users' lives. Of course, it will also have some biases, which are based on the biases of the corpus writers themselves. Such as: 'In some large corpora, robots will think that doctors correspond to men and nurses correspond to women', etc.. I think the corpus can be edited by more people

with different characteristics, delete the biased data set and solve these biases, and finally get a relatively fair corpus for application. In addition, some algorithms to eliminate bias can also be used to delete some biased correspondences, thereby eliminating the bias of the robot itself.

Conclusion:

In this work of mine, I learned how to configure the environment, document conversion, function calls, deep learning, model training and adjustment parameters, some basic principles and usage methods of nlp, and some basic methods of web crawling. These make my chat bot gains more functions. Through user tests, it can achieve the expected goals of chat, question and answer, query date, query weather and playing games. I think this is a relatively successful work.