TAL Report

CHEN Hang
FAN Zheng
XU Li

May 2018

1 Description

The goal of our project is to make a three-modes, English speaking chatbot, whose general function is to have a simple daily conversation with an user, and whose special function is to help users to search for the information about the matches of NBA.

There are three modes for our chatbot "JJ":

Mode 1:

- "JJ" responds to the user in a negative manner with the usual words, for example, *"uh huh"*, but each time the response will be different from the previous.

Mode 2:

- "JJ" can detect the sentences like *"i am X"*, *"he is X"* ... or their abbreviations, such as *"i'm X"*, *"he's X"* ... Then "JJ" will make corresponding changes to subject and verb, and make the answer, like *"Why are you X?"*
- "JJ" can identify some keywords and ask the user a question about the topic corresponding to the keywords. Each question mentioned will be different from the last question.
- If "JJ" cannot find the response, it will degrade to mode 1 automatically.

Mode 3:

- "JJ" becomes the assistant of NBA(the National Basketball Association of America). In this mode, "JJ" has 4 principal functions:
  - Search for the information of NBA players(He can tell us about more than 600 players' information in NBA);
  - Search for the information of all 30 teams of NBA;
  - Show the schedule of certain day;
  - Show the score between two teams of certain day.

- If "JJ" can't understand the question about NBA, it will reply with a general answer, which we have trained by the model *"neural machine translation"*.It needs to download the *"model"* folder and do what I have mentioned in the README. If not, "JJ" will answer with mode 2 or mode 1.

2 Motivation

We have two motivations for this chatbot.

- For the ordinary functions of this chatbot, it can have a daily dialogue with our users, and it is able to listen to the happiness of users, relieve the troubles and loneliness for our users.
- As for the special function of this chatbot, it can provide basketball enthusiasts with sport events, the information of teams, and players, which can bring the convenience to users. What's more, the chatbot will be used to analyze the result of the score or to predict the winner of the match with the development.

3 Future Plan

- As for the general functions, we hope to develop our chatbot to be like a good friend of our users, who will be able to provide users with psychological counseling and provide helpful suggestions on the difficulties they face. We also hope that according to the user's choice, it can identify different languages.

  We will add *nltk* library in this section to make it possible to analyze the sentences entered by users well, so as to increase the recognition of complex sentences and make "JJ" respond to users better.
- As for the specified functions, the chatbot is not only a specialist of NBA, but also a expert in other sports like football, tennis and so on. Its powerful function will attract more and more people who like sports. We will think about putting different modes in the chatbot. For example, there will be some quizzes of sport when the young people chat with it. And for some people, the chatbot will provide them with the information according to their interests.
- As for the aspect of business, we can cooperate with some sport companies. Sometimes the chatbot will talk about some advertising in the daily chatting with users. What's more, if the company of collaboration has some products to sell, users can also buy them directly in the chatbot.

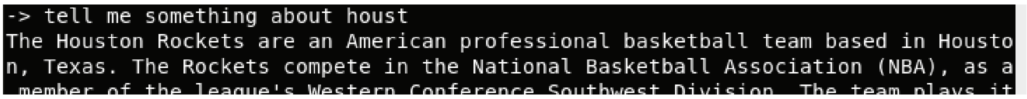
4 Realization of Chatbot

- For the implementation of structure of dialogue, we use an infinite loop. We use *re.match()*to verify the sentences. If and only if the sentence entered by users is *"exit"*, the conversation ends. We use the variable *"mode"* entered by users to determine which mode will be called.
- In addition, we create *"global_init.py"*, which is used to ensure that variable *"mode"* can be used as a global variable for all *x.py*.

- As for mode 1, we use the function *random.choice()* to randomly select a response in *BACKCHANNELS*, which has already enumerated six responses, and create a global variable *"record"* to record the previous response in order to avoid repetition of the last one.
- As for mode 2, users enter sentences passed as parameters. First of all, we create global variables to save the keywords, the questions for each topic, and *"numero"* to record the number of sentence used last time.
  - As for the sentence such as *"i am X"*, at first, we use function *re.search()* to verify if there are keywords such as *"i am"* in the sentence. Then we use *re.split()* to split the structure of the sentence and extract the part X and the part subject. If there are some specific subjects, the corresponding changes of the subject and the verb are performed; otherwise, the subject is treated as he. Finally "JJ" respond a sentence with *why + verb + subject + X ?*
  - As for Asking questions on corresponding topic according to the keywords, we use *re.search()* to determine if there is a keyword in the sentence. If so, a problem random number will be generated, and "JJ" will respond users according to the number. Otherwise, it prove that the "JJ" can not answer the user's words, "JJ" will call a method *global_init.minus_mode()* to automatically degrade to mode 1.
- How to realize these 4 functions of mode 3.
  - For understanding the request that user input, we use the *NLTK* to tokenize the sentences. And we do the *POS* tagging for each word. For the initialization, we build the *BACKCHANNELS* for players, teams and some key words. For the *BACKCHANNELS* of players, we build the list of players with the function *'init_player()'* in *"nba_search/playerinit.py"*. It will return a *'set()' 'player_list'* who contains those names of NBA players. We choose 2 websites to catch the names of NBA players, so there are 2 parts of operation with the html of the website.
  - After building the *BACKCHANNELS*, we classify the words by noun, prep, adj, date...

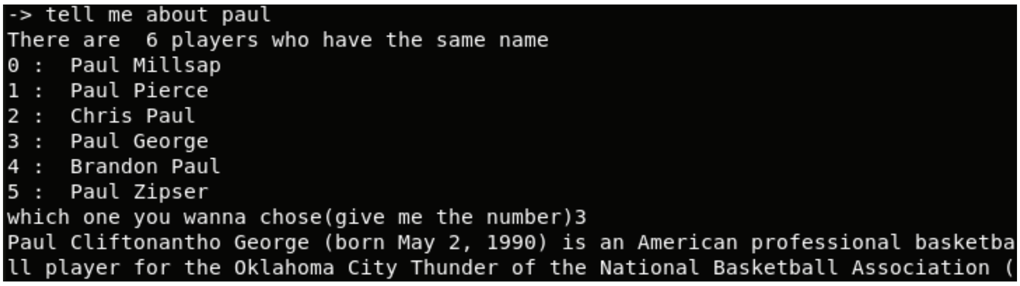
    Then we match the nouns with the *BACKCHANNELS* to understand the meaning of the request. For example, if someone want to search for the score of two teams, the nouns will match the *SCORECHANNELS*, so we know what he wants. And for the *teamchannel* and *playerchannel*, we use the fuzzy query. Users don't need to input the exact word in the set or list or dictionary.

    For example, you want to know something about Houston Rocket, you just need to input *"houst"* or *"rock"* that will be fine.



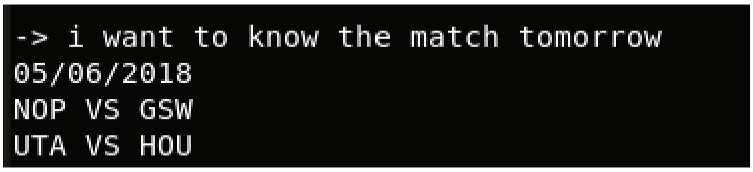
```
-> tell me something about houst
The Houston Rockets are an American professional basketball team based in Housto
n, Texas. The Rockets compete in the National Basketball Association (NBA), as a
 member of the league's Western Conference Southwest Division. The team plays it
```

  - For the players who might have the same name, the chatbot will reply some options with all the names of player you are interested in.



```
-> tell me about paul
There are  6 players who have the same name
0 :  Paul Millsap
1 :  Paul Pierce
2 :  Chris Paul
3 :  Paul George
4 :  Brandon Paul
5 :  Paul Zipser
which one you wanna chose(give me the number)3
Paul Cliftonantho George (born May 2, 1990) is an American professional basketba
ll player for the Oklahoma City Thunder of the National Basketball Association (
```

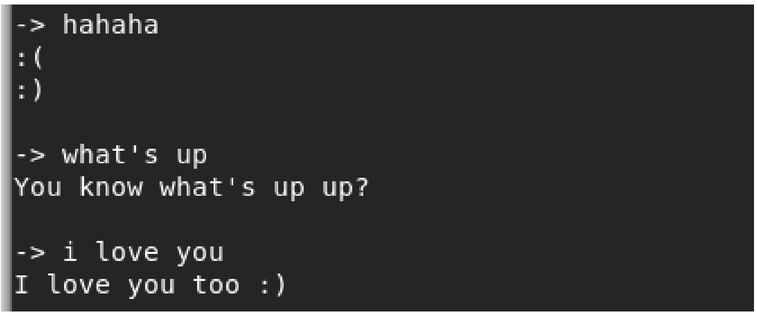
- When users input the date, they need to obey the rule *”mm/dd/yyyy”* or they can use the word like *”today”*, *”yesterday”*,*”tomorrow”*.



```
-> i want to know the match tomorrow
05/06/2018
NOP VS GSW
UTA VS HOU
```

All the functions for searching the information in the internet is in the folder *”nba_search”*. We use the Web Scraping in the web of Wikipedia for the information of player and teams. We catch the useful information of html in the website and use regular expression to get the introduction that we need.

- And for the question that is not referred to the topic of NBA, we use the model “neural machine translation” for training the chatbot. This model is based on the theory called “seq2seq”. We have learned from the tutorial in github. (https://github.com/tensorflow/nmt) Even though it is a model for translation, but it depends on the corpus we use. If we use the corpus for chatting, it will work here. So first, we downloaded the month comment of *reddit* in the site: *http://files.pushshift.io/reddit/comments/*. We chose the comments of 2017-11. And we write the *db_create.py* to build the database and record all the topic and comment whose score greater than 2. Maybe there are some topics with many different comments, we always update with a better score, so in the last we choose the highest score one which we think normally is the best. After that, we write the *train_create.py* to build the train set. Then we have a long time for training. The following image will show the result.

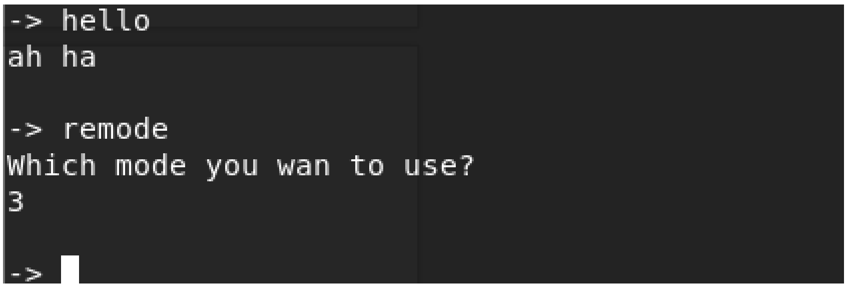


```
-> hahaha
:(
:)

-> what's up
You know what's up up?

-> i love you
I love you too :)
```

When you find the mode is degraded, you can input *"remode"* to re-choose the mode you want. We use a *re.match()* after the *input()* in *main.py* to check the word *"remode"*, which means that a user want to change the mode.



```
-> hello
ah ha

-> remode
Which mode you wan to use?
3

->
```

5 Limitations of the project

- For the mode 2:
  - "JJ" can't really understand the meaning of the user's sentence, but responds mechanically to the recognized sentence pattern.
  - Recognition for some complex sentences is not high enough, for example, for the type of sentence *"i am X"*, a user enter *"i am so so so happy and i want to go to the park."*, if there is no punctuation marks behind the part X, the response will be *"why are you so so so happy and go to the park?"*

- For the mode 3:
  - The chatbot can only do the search online, so the user can't use the mode 3 without connecting the internet.
  - The list of player is limited, we have recorded about 600 players which contains 100 most famous players and 500 players who play in NBA currently. If the user want to search for the information of other players, the chatbot won't show the result.
  - For the function of searching the score, the chatbot can't get the result within 2 days(yesterday, today and the future) because the site web where we do Web Scraping doesn't have the information in the html.
  - When the user want to search the information about certain day, he needs to input the date with the form of *"month/day/year"*(like 05/05/2018).
  - We have not used all the comments of *reddit* in 2017-11 because it needs much time to deal with the data and training. So the answers sometimes are not very suitable. And there are many response with url like *www.youtube.com..* ,as you know it is the result of using the comment of a social network.

6 Division of work

XU Li and CHEN Hang: They are responsible for the completion of mode 3.(question understanding, web scraping, database and trainset building, training with model *nmt*)
FAN Zheng: She is responsible for the completion of mode 2, a small part of mode 3 and The construction of structure of dialogue.