The Baidu-Tieba chatbot based on Rasa-nlu

Yunzhe Li

Chapter 1 Project Background

1.chatbot

Chatbots are software applications which are designed to conduct online chat conversations via text or text-to-voice rather than directly contacting with a human agent. Chatbot systems are designed to convincingly mimic how humans behave as conversation partners typically require constant tuning and testing, and many products still fail to perform adequate conversations or passing the industry-standard Turing test. The term "Chatter Bot" was originally coined by Michael Mauldin(creator of the first Verbot) in 1994 to describe these conversational programs.

Chatbots are used in conversation systems for a variety of purposes, including customer service, request routing, or information gathering. While some chatbot applications use extensive word-sorting processes, natural language processors, and sophisticated artificial intelligence, while others simply scan for generic keywords and generate responses using generic phrases retrieved from relevant libraries or databases.

Most chatbots are accessed to online via web site pop-ups or virtual assistants. They can be categorized according to usage categories. It includes these things: business (e-commerce via chat), education, entertainment, finance, health, news, and productivity.

2.Baidu-Tieba

Tieba, which is also known as Baidu Tieba, is an independent brand of Baidu and the largest Chinese language community in the world. Tieba was the brainchild of Robin Li, who is the CEO of Baidu. His aim is to build an online communication platform combined with the search engine, where people with the same interested topic can easily communicate and help each other. Tieba is a topic exchange community based on keyword, which is closely combined with search. It can accurately grasp the needs of users. All in all, it is born for interest.

The mission of Tieba is to bring together those like-minded people. Tieba can accurately gather a large number of friends, make them able to show their own style, and help them to make friends. It can also build a unique interactive platform of "interest theme" depending on search engine keywords.

With the changes of the Internet environment, the clients of Tieba are becoming bloated, and non-super members will receive a large number of advertisements when using Tieba. The high frequency of advertisements has exerted a certain influence on the use of Tieba. Figure 1 shows two relevant screenshots of Tieba clients. This design attempts to create a mini tiebar client by making a dialogue robot, and it also achieve some other dialogue functions. I am trying to find another way to open the post bar. All in all, it will make the surfing more easily. It will also avoid the advertisements.

Figure 1. Current Tieba client screenshot

Chapter 2 design and implementation

1.algorithm flow

The main process of this design is: setting some states through the idea of finite automata, so as to realize the realization of the operation from the beginning, entering the Baidu Post bar, entering the post, turning the page and return. It can also avoid changing the state when the user`s input has no obvious intentions.

2.Data and Framework

For the data set, I used the Chinese data set of SMP2019(the 8th National Social Media Processing Conference). It is used in the chat part of the design, which is used to support the intention recognition of a certain degree of query functions, such as the route query, location query, etc. In addition, most of the data sets used in this design are procedually-generated, which are used to support semantic understanding of tieba function. For example, the understanding of different expressions such as "the first post", "Please enter the first post", "I want to see the penultimate post" and so on. They should be supported to enter the post function. It has the ability to recognize the requirements of transfer to post bar dialogue data generation for all types of post bar names, such as "I want to go to the pressure back pot bar", "I want to go to the post of Warcraft" and so on; The realization of these functions requires a large amount of data support. Besides, the data itself also needs to contain entities and values to support entity detection, which needs to be obtained through automatic generation.

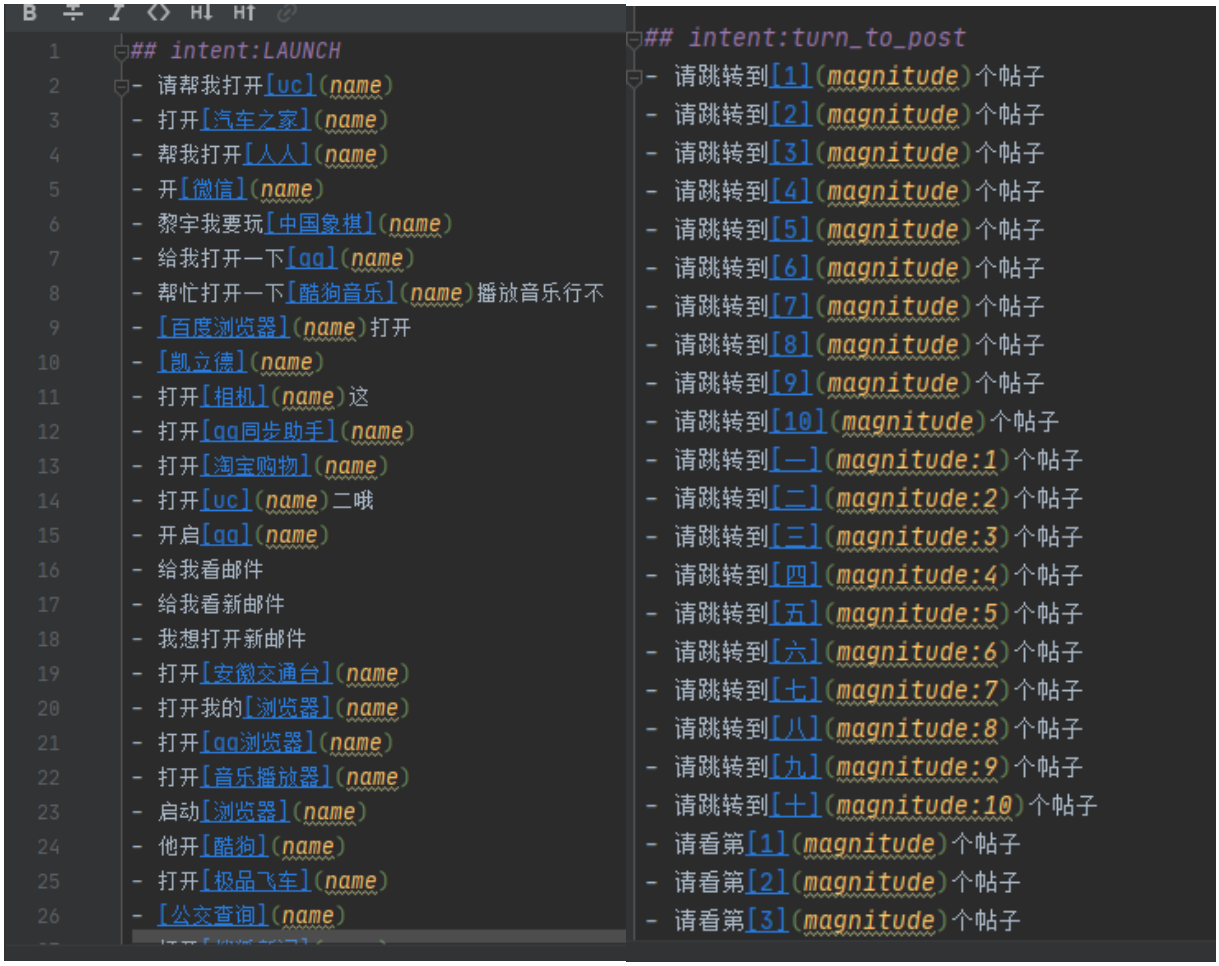


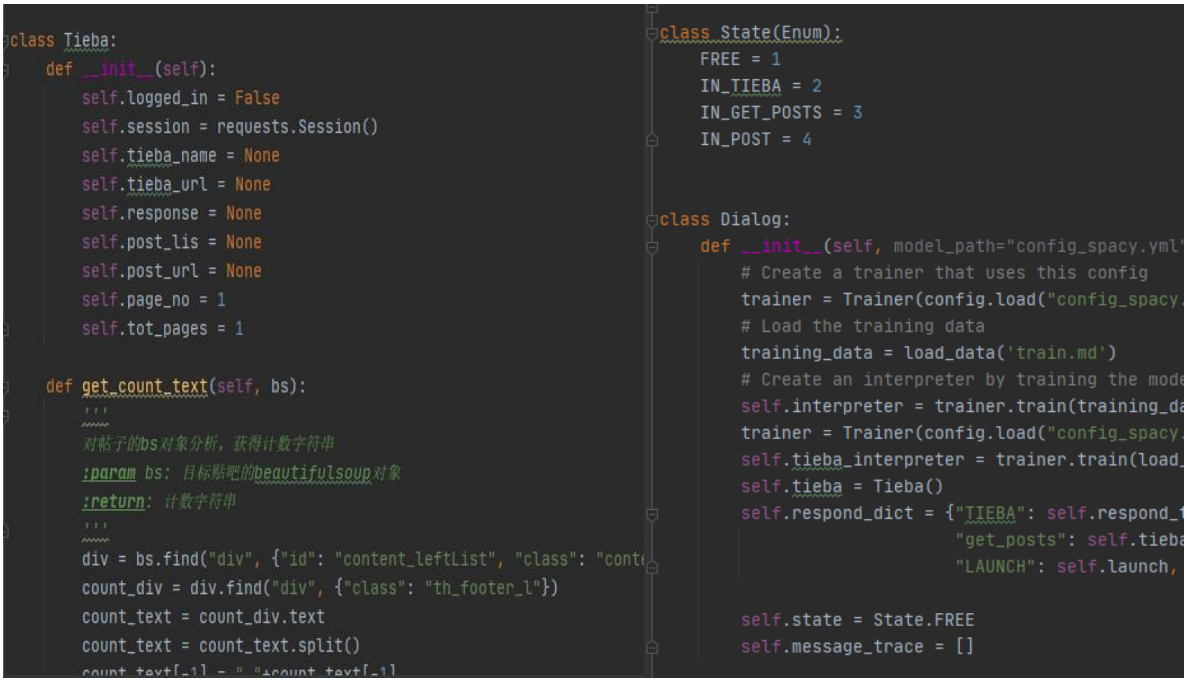
Figure 2: Partial data set; Part of the SMP2019 dataset is on the left side and part of the self-generated dataset is on the right side

For the framework, this design is completed on Python, mainly based on rasa-nlu library. In addition, many other third-party libraries are used in it: crawler, regular matching, CHAT software API call and other related requirements.

The table 1 shows some versions of third-party libraries involved in this design.

library	versions
cn2an	0.5.5
telegram	0.0.1
requests	2.24.0
rasa_nlu	0.15.1
beautifulsoup4	4.9.1

Table 1: Examples of third-party library versions used



```
class Tieba:
    def __init__(self):
        self.logged_in = False
        self.session = requests.Session()
        self.tieba_name = None
        self.tieba_url = None
        self.response = None
        self.post_lis = None
        self.post_url = None
        self.page_no = 1
        self.tot_pages = 1

    def get_count_text(self, bs):
        '''
        对帖子的bs对象分析，获得计数字符串
        :param bs: 目标贴吧的beautifulsoup对象
        :return: 计数字符串
        '''
        div = bs.find("div", {"id": "content_leftList", "class": "conte
        count_div = div.find("div", {"class": "th_footer_l"})
        count_text = count_div.text
        count_text = count_text.split()
```

```
class State(Enum):
    FREE = 1
    IN_TIEBA = 2
    IN_GET_POSTS = 3
    IN_POST = 4


class Dialog:
    def __init__(self, model_path="config_spacy.yml
        # Create a trainer that uses this config
        trainer = Trainer(config.load("config_spacy
        # Load the training data
        training_data = load_data('train.md')
        # Create an interpreter by training the mode
        self.interpreter = trainer.train(training_da
        trainer = Trainer(config.load("config_spacy
        self.tieba_interpreter = trainer.train(load_
        self.tieba = Tieba()
        self.respond_dict = {"TIEBA": self.respond_t
                             "get_posts": self.tieba
                             "LAUNCH": self.launch,

        self.state = State.FREE
        self.message_trace = []
```

- **Figure 3:** Snippets of code for major features

3.concrete realization

For the implementation. In addition to the code for the temporary generation of data sets, and the main file which is used for the realization of the overall process. This design is also mainly used through the implementation of the following classes to achieve the goal： Tieba、Dialog for chatting，and State for enumeration. The Tieba class mainly relies on crawler technology to achieve it. and getting part of the implementation of the construction of mini Tieba client through the HTML analysis of

Tieba web page end. The Dialog class, which is closely related to the State enumeration class, using finite automata by maintaining a dictionary from (State, input) to (State, operation function). In the specific dialogue robot platform. As the wechat platform itChat, QQ platform CoolQ and other related API has been disabled (CoolQ was disabled within a month). I will use Telegram API to achieve in this design.

Chapter 3 Function display

The dialogue robot implemented in this design can perform basic query operations to a certain extent, including location query, recipe query, route query, etc.

Figure 4 Basic query function

The post bar`s function which is realized by this design can effectively support the reading operation of the post bar. It includes entering the post bar, entering the post, turning pages in the post, etc. By maintaining the operation stack, we can realize the return of the previous step.

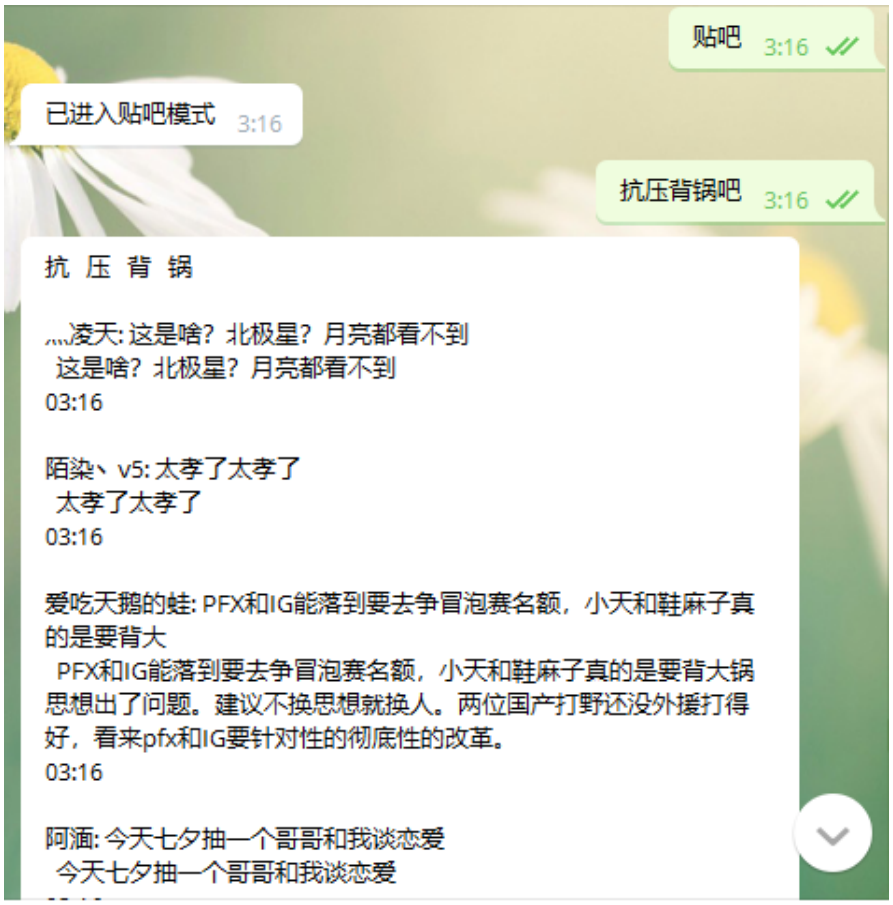


Figure 5 Post bar browsing function

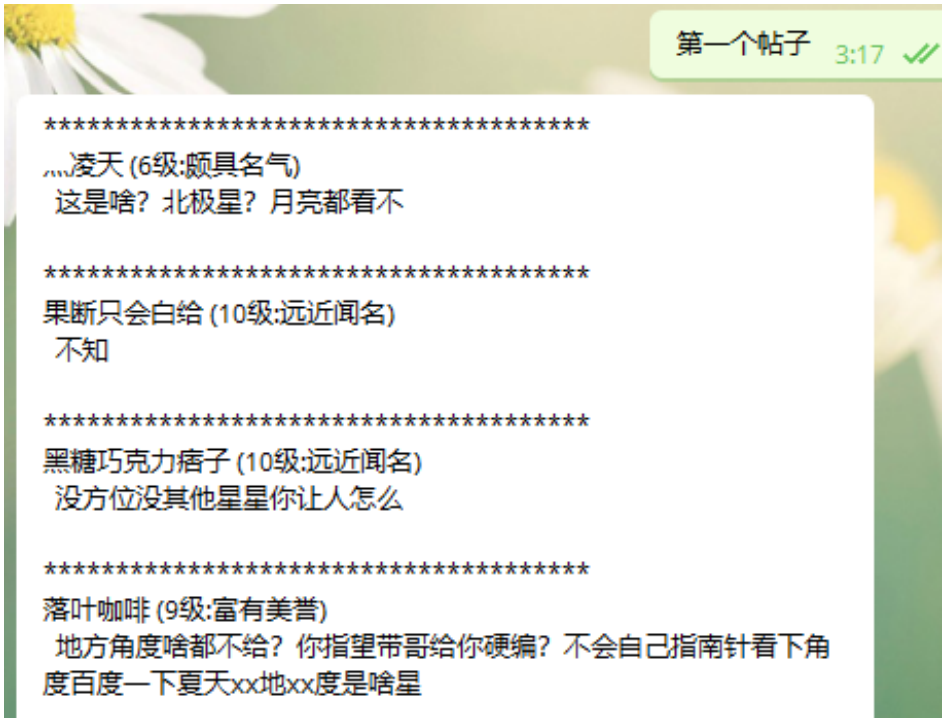


Figure 6 Post browsing function

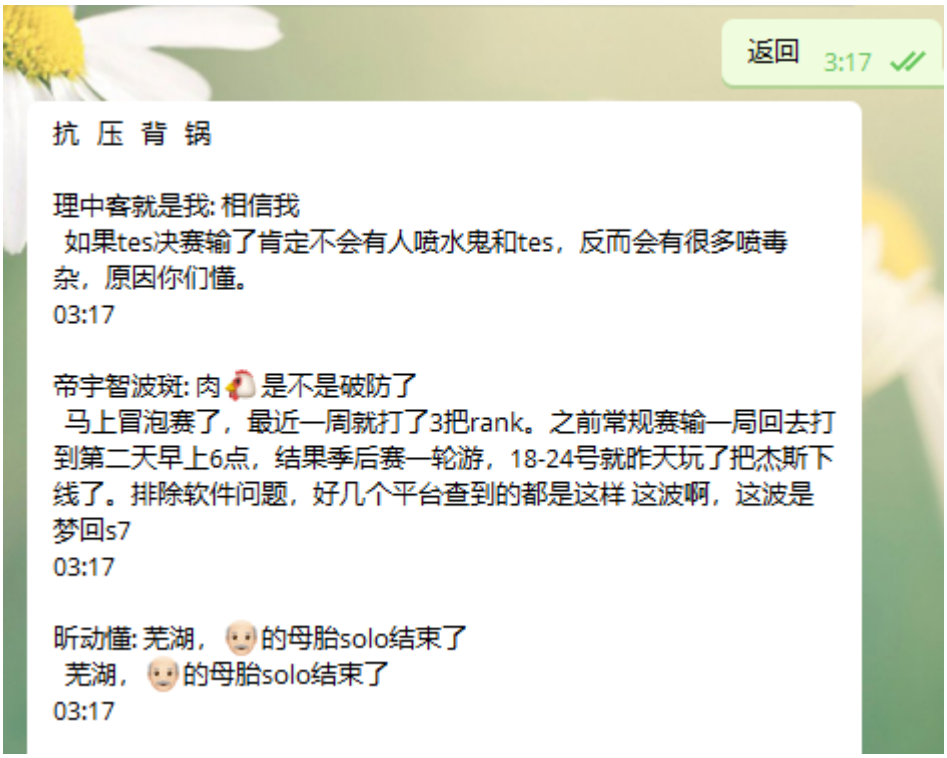


Figure 7 the function of return

Chapter 4 summarizes and deficiencies

I have realized a dialogue robot with certain basic functions and post bar reading functions. It tries to solve the problem of tieba client which is becoming bloated. However, due to the complexity of Baidu login system restrictions and other reasons. This design does not implement the anticipated account login management through Requests sessions. I want to try to solve these problems in the near future.

This is my first time trying to acknowledge the AI, I know what I have done is easy. I hope I can get the graduate studies in the future and do something about the artificial intelligence.