

University of Nottingham

Human-AI Interaction (COMP3074 UNUK)

AN INTERACTIVE NLP-BASED AI SYSTEM

-----Dongchen Yao – psxdy2 -

20349894

Content

University of Nottingham	1
Human-AI Interaction (COMP3074 UNUK).....	1
AN INTERACTIVE NLP-BASED AI SYSTEM.....	1
1. Introduction	2
2. Background	3
3. Proposed system.....	5
a.Functionality.....	5
b.Originality	5
c. Implementation	6
4. Evaluation.....	9
5. Discussion	13

6. Conclusion.....	14
7. Reference	15

1. Introduction

The idea of constructing my psxdy2 chatbot system comes from the six potential features of chatbots given in the documentation task, and making an intelligent chatbot and covering Intent matching, Identity management, Small talk, Transactions, Information retrieval & question answering, and game-playing functions, in general, need to have a conversational goal, be able to generate contextually coherent users, and have sufficient generalization capabilities to generate reasonable behavior in the conversational situations that arise in the corpus.

So I chose to self-produce based on the Rasa framework because designing chatbots by writing rules manually is costly despite the handedness and high accuracy rate, so seeking a data-driven modeling approach is a good way to go. End-to-end training using conversation corpus is better than the rule-based approach, which has the advantage of being data-driven and efficient; however, the disadvantage is that it is difficult to model complex conversations and requires a high amount of data, so it is ineffective for some domains where conversation

corpus is scarce.

In the following, I will describe the background of the chatbot production, the project process, and the inspiration of the production, as well as explain the technology of the system, the originality of the project, and the implemented functionality. The implementation of the system will be further evaluated and a user test script will be given. Finally, a discussion and conclusion will be given based on the above.

2. Background

In the production of which I referenced the rasa model and open-source code, while a qualified chatbot must also have a refreshing UI front-end interaction interface, which has prompting options to allow users to better make an opening, this is my background preparation work in the pre-production.

Rasa Code : <https://github.com/RasaHQ/rasa>

Rasa Reference Models : https://github.com/Dustyposa/rasa_ch_faq

Two interfaces are used

CITY_LOOKUP_URL = <https://geoapi.qweather.com/v2/city/lookup>

WEATHER_URL = <https://devapi.qweather.com/v7/weather/now>

Using Rasa, a human language can be analyzed for text, and the main result of the analysis is the intent and entity information. And the entity information is used for slot filling in the dialogue system. Also, the intention and entity information obtained from the analysis in the conversation management is used for slot filling, and when combined with the state of the previous rounds of conversation, the policy model obtained from the training based on deep learning, reinforcement learning, decides how the current user's conversation should be responded to. Then according to the chatbots that exist now inspire me is the core technology of natural language processing in the semantic interaction level of chatbots. For example, retrieval techniques can select the most appropriate responses in the corpus, named entity recognition can identify the key information in the sentence and be used in combination to parse the user's sentence.

And using database technology we can quickly retrieve similar sentences in a pre-stored large scale corpus and store and further analyze them like in recording name information here when the user gives my name is X, then all subsequent ones will have Hello X as a prefix, whether you are in a transaction or in casual conversation, which will also give the user a great sense of closeness and intelligence.

3. Proposed system

a.Functionality

1. Task completion type (used to complete the user's specific task needs, such as airline ticket booking, etc.): **city weather matching** and **name information storage** and **transaction** in the system.

2. Question and answer type (used to solve the user's information query needs, question and answer type is mainly a question and answer and customized question and answer, for example, to implant the robot question and answer, the user can get the robot back according to the fixed corpus trigger, mainly in the form of a question and answer dialogue, such as "how is the weather today", etc.): **Information retrieval & question answering, Game playing** in the system

3. Type of talk (used to solve the user's emotional needs and other types of dialogue between the interface needs, at present, the main retrieval and generation of two options: **Small talk**

b.Originality

In order to make the bot appear smarter and more personalized, thus enhancing user satisfaction with the product, I crawled the web for corpus review as well as

manual writing. And I collected a lot of intellectual quizzes or sentences with meaning from the Internet that can bring surprise to users. After giving many seed users experience, they think the robot is interesting as well as a novel, beyond expectation, and also give a real-time weather system for the city of the transaction, which is a new experience breakthrough for all users, and get unanimous praise.

c. Implementation

1. Rasa modifications to add corpus and answers.

(a). Add corpus: add corpus in data/nlu.yml file.

(b). modify the corresponding answer: find the purpose of the answer that needs to be modified from nlu.yml, go to the corresponding in domain.yml, modify the answer

2, Rasa to increase the intent (single round, more complex multi-round, refer to the official website)

(a). In data/nlu.yml add new intentions and corresponding materials

(b). Add new intent and corresponding pipeline in data/rules.yml (single round)

(c) . Newly add the name of the answer in domain.yml and the corresponding answer to the answer

Deliverable 1 - NLP system [20211125] C:\Users\Administrati

formbot_en

actions

data

game_data

query.txt

response.txt

utils

init.py

coins.py

request.py

search.py

actions.py

data

nlu.yml

rules.yml

stories.yml

models

config.yml

domain.yml

endpoints.yml

formbot.log

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

```
version: "2.0"

nlu:
  - intent: greet
    examples: |
      - Hi
      - Hey
      - Hi bot
      - Hey bot
      - Hello
      - Good morning
      - hi again
      - hi folks
      - hi Mister
      - hi pal!
      - hi there
      - greetings
      - hello everybody
      - hello is anybody there
      - hello robot
```

Deliverable 1 - NLP system [20211125] C:\Users\Administrati

formbot_en

actions

data

game_data

query.txt

response.txt

utils

init.py

coins.py

request.py

search.py

actions.py

data

nlu.yml

rules.yml

stories.yml

models

config.yml

domain.yml

endpoints.yml

formbot.log

rasa_bot_front

README.txt

External Libraries

Scratches and Consoles

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

```
version: "2.0"

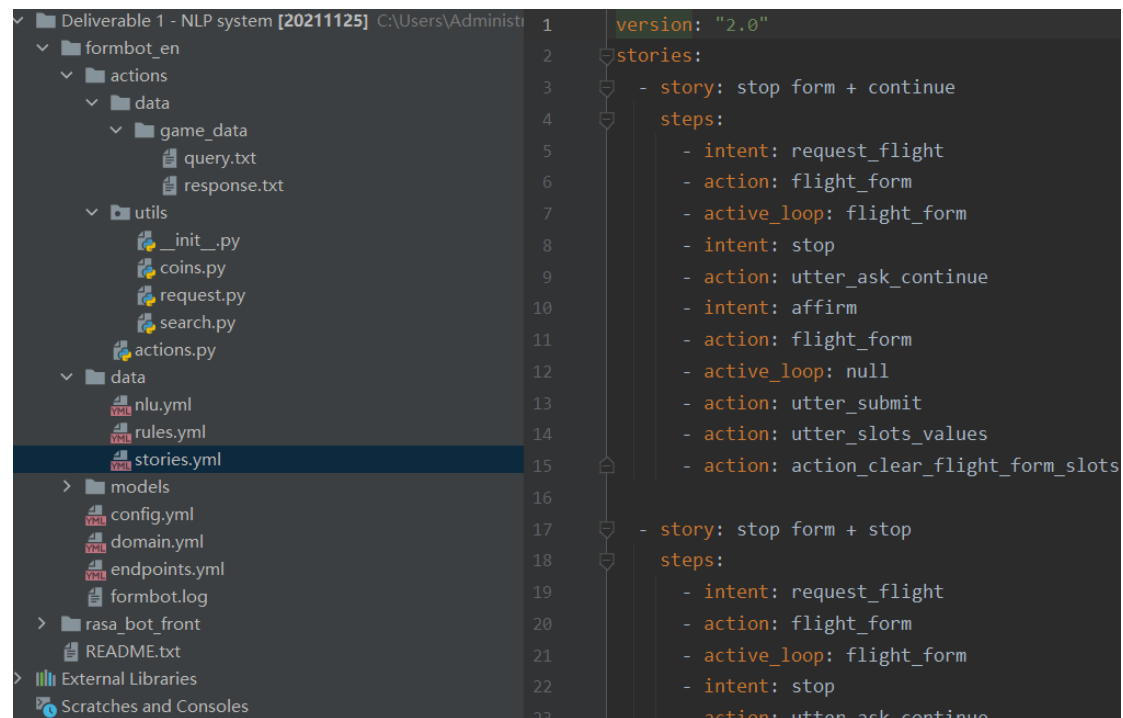
rules:
  - rule: Greet user
    steps:
      - intent: greet
      - action: utter_greet

  - rule: Thank you
    steps:
      - intent: thankyou
      - action: utter_noworries

  - rule: Bot challenge
    steps:
      - intent: bot_challenge
      - action: utter_iamabot

  - rule: weather_search
    steps:
      - intent: weather_search
      - action: action_query_weather

  - rule: ask_again
    steps:
      - intent: ask_again
```



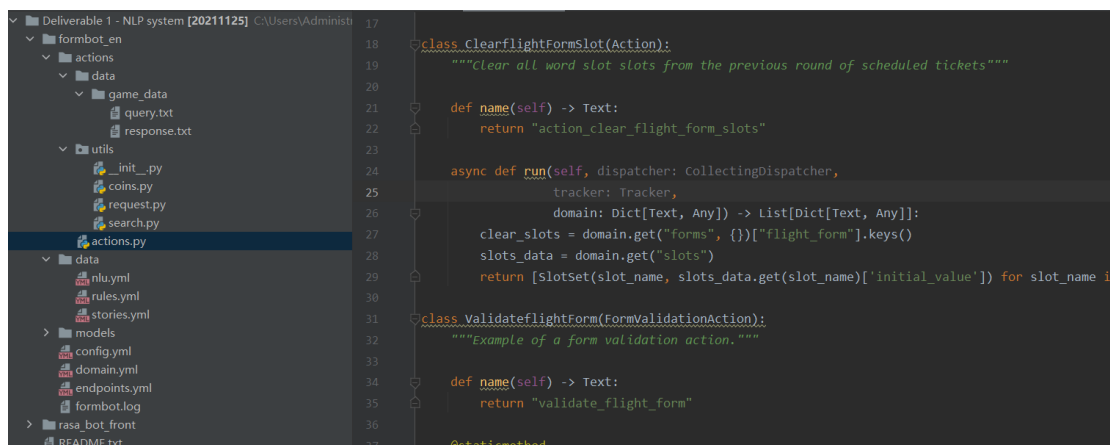
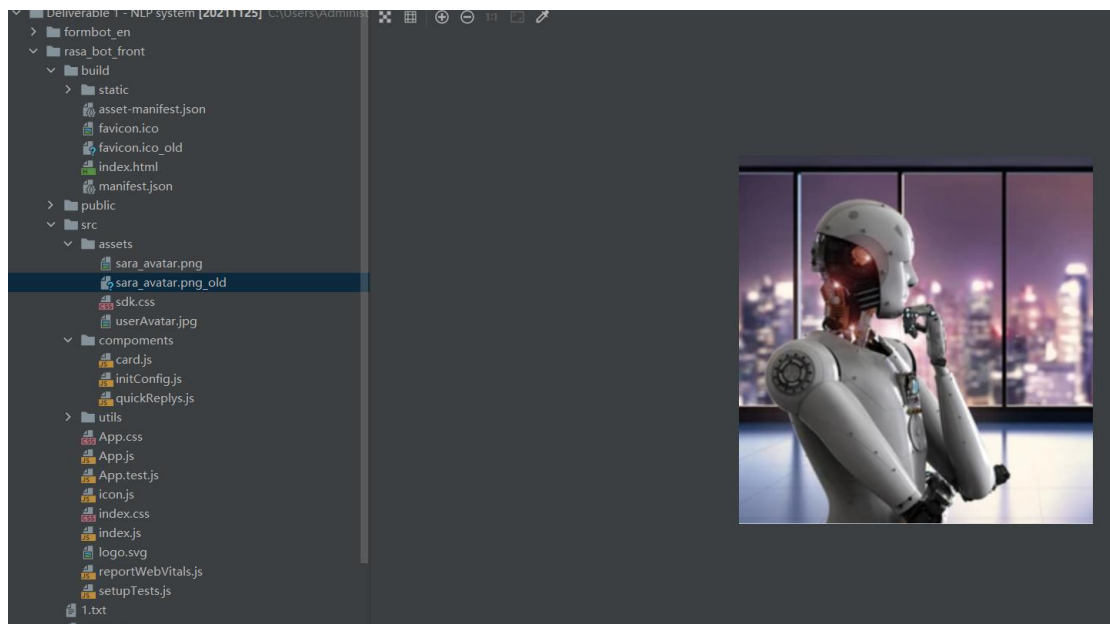
3. Rasa training commands.

(a). First backup the tarball in models

(b). To the domain to establish a directory, the implementation of Rasa train

(c) If no error is reported, the model will be modified to generate the latest tarball after the training. If the model is wrong, modify the model according to the log

4. At the same time, I also designed the front end, so that users can have a better user experience of chatbot, and also laid a good foundation for my subsequent secondary development.. See README for details.



4. Evaluation

I give user scripts from the following aspects to test the project

(1) Good semantic criteria

Ability to understand the questioner's intent (to understand human intent)

Answers that match the question (make correct and appropriate feedback)

(2) Design semantics

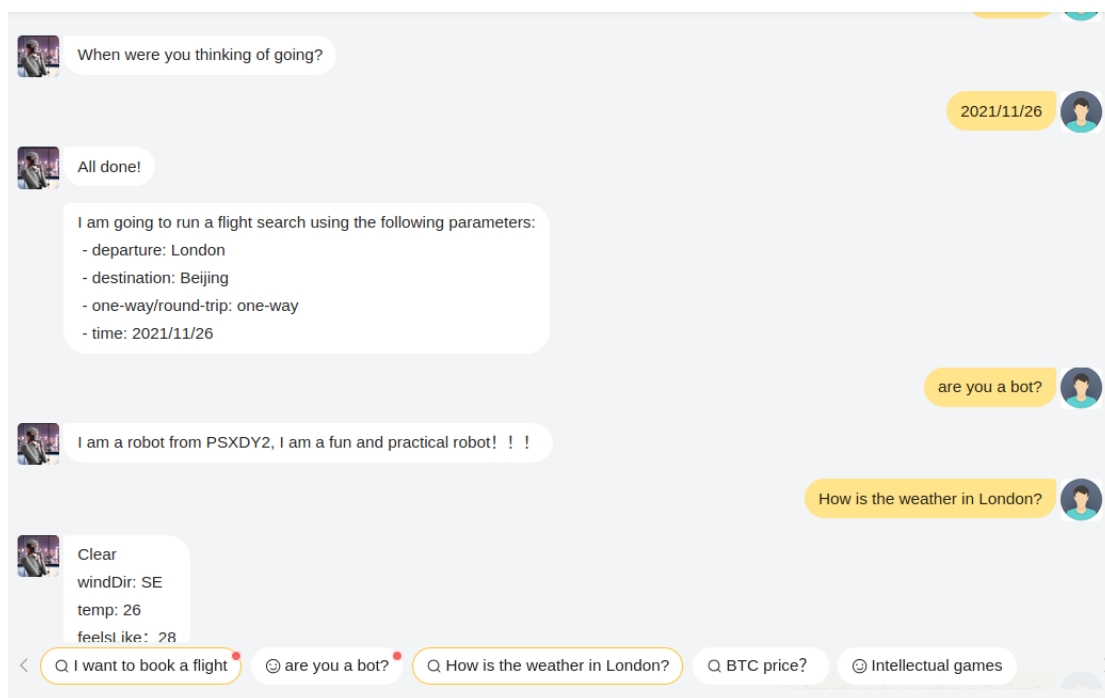
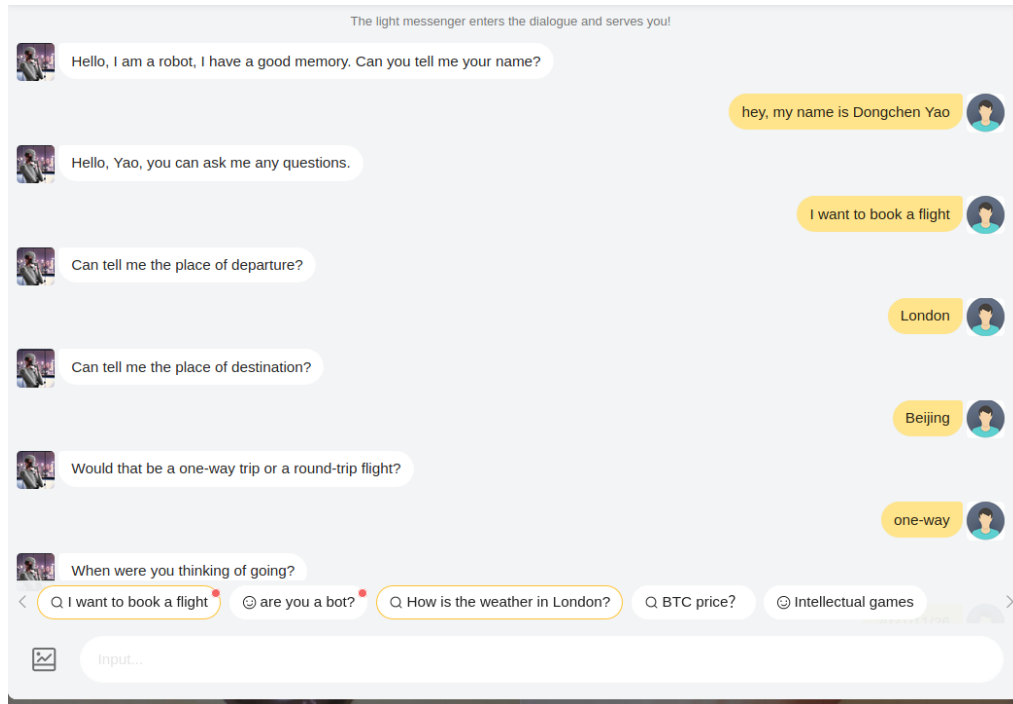
The answer is as personalized and flexible as possible and slightly guides the user.

For example, "You can ask me about booking tickets"; when you can't reply to a user, you need to use short language to let them know you can't answer with limited ability.

Increase the active interaction scenarios of the robot, remember the user's name, and improve the interactivity while meeting the user's demands.

(3) User needs

- Help users and go to solve the current problems encountered
- Transactions or intent matching to achieve accurate results
- When the robot does not recognize something within the corpus, it can give the user feedback tips



How is the weather in London?

Clear
windDir: SE
temp: 26
feelsLike: 28

what about the weather in Beijing?

Rain
windDir: SE
temp: 11
feelsLike: 3

thank you

You are welcome :)

i wanna know something about BTC

Q I want to book a flight

are you a bot?

Q How is the weather in London?

Q BTC price?

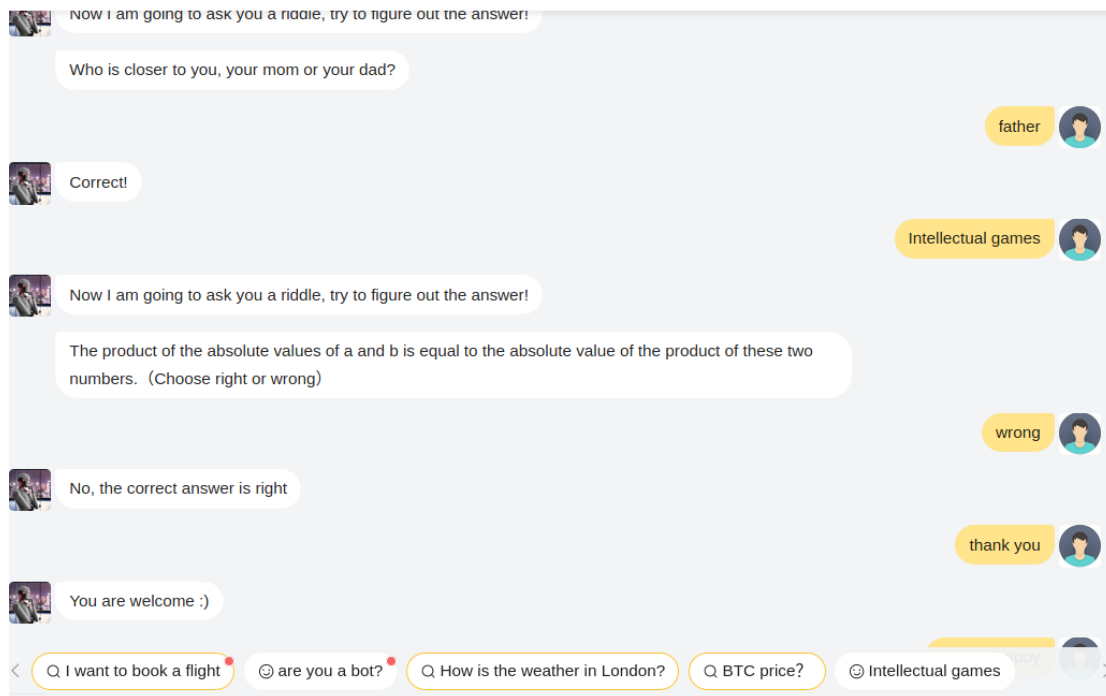
Intellectual games

i wanna know something about BTC

ok, let is play the game

Now I am going to ask you a riddle, try to figure out the answer!

What clothing is always sad ?



5. Discussion

If my bot system is deployed in the real world, since more users are involved, it will allow the bot to do better deep learning and improve the corpus, first I will improve the content of the data set more, the technology is not in mature stage, to make a chatbot that is slightly satisfactory, the most critical is the manual, manual is greater than the data, more than the algorithm factors.

Then in the real system, chatbots will be more inclined to a product-oriented design. Since chatbots are limited by the technology that cannot meet human expectations, then whether it is possible to make users not focus on the technical performance itself from the perspective of product design, but to generate stickiness to the product from other dimensions, this is a kind of thinking I have for my system to be

realized in the future. The key to making products is to exceed user expectations, which is obviously not friendly to AI products because user expectations are too high, so it is necessary to think in other dimensions, such as product strategy: let the chatbot have a sense of life, let it be personified and IP-based, so that users will not only focus on the dialogue, but will generate stickiness to the product and better user experience from more demand levels, and subsequently, If I continue this project, I will not only make the system more human-like in intelligence but also increase the intimacy with people in terms of IP or UI interaction to increase user stickiness, while the increase in volume and corpus correctness will also prompt a virtuous cycle of learning for chatbots.

6. Conclusion

In this system of making chatbots, it gave me a great challenge to go from training corpus to single-round conversations to multi-round conversations, and then to train a good model, based on setting intention classification and other Rasa algorithms, to train, and when the bot could meet my expectations based on the user's answers, after trading and small talk one at a time, it was a sense of accomplishment as if creating a new life. And at the same time, I realized the kind of machine learning, AI integration that can overall make the user have a revolutionary help and better experience.

7. Reference

Rasa Code : <https://github.com/RasaHQ/rasa>

Rasa Reference Models : https://github.com/Dustyposa/rasa_ch_faq

Two interfaces are used

CITY_LOOKUP_URL = <https://geoapi.qweather.com/v2/city/lookup>

WEATHER_URL = <https://devapi.qweather.com/v7/weather/now>