STOCK-CHATBOT PROJECT REPROT

Wei Zhihang

2019-7-7

ABSTRACT

In this project, we built a WeChat chatbot based on RASA natural language understanding framework. The chatbot is applied in the field of stock information query, which has three functions: current stock query, historical stock query and stock analysis.

The main techniques used in this project include: using pattern matching, keyword extraction, and syntactic transformation to answer questions; Extracting user intention by the nearest neighbor classification; Named entity identification by package spaCy; Implement multi - round query technique based on state machine.

In addition, the chatbot uses wxpy library as the API for WeChat access and iexfinance library as the API for stock query function.

Finally, after tested, the robot is verified to have the basic chat response function, and have a good effect on the stock information query function.

Key words: artificial intelligence; natural language understanding; chatbot;

CONTENT

1 INTRODUCTION

When technology research can be put into practice, the demand for it suddenly increases. From the treatment of talents in the field of artificial intelligence by major universities and companies, we can know the great value brought by this technology. In all aspects of life, artificial intelligence is unconsciously integrated into our lives. The popularity of automatic driving, face recognition, speech recognition, and artificial intelligence chatbot has long exceeded people's imagination.

Chatbots can be used for different purposes in many fields. Now when we visit websites, when we use devices (Google Assistant, Apple Siri, Microsoft Cortana), or when we use instant messaging applications, we can find chatbots.More and more teams are launching chatbot platforms or apis, such as Facebook, Telegram and WeChat.Many companies, such as e-commerce companies, Banks and hotels, can use chatbots to answer simple questions to avoid unnecessary duplication and improve efficiency.

Chatbots have broad application prospects, so I took this course to implement a simple chatbot, whose purpose is to provide people with stock and weather information. This report mainly introduces the working principle of the chatbot and how to implement it.

2 THE METHOD

2.1 Intent Extraction

Intent extraction is a type of Natural-Language-Understanding (NLU) task that helps to understand the type of action conveyed in the sentences and all its participating parts.

2.2.3 Regular expression

Regular expressions are used to match strings, find strings of a certain format, extract keywords, such as phone Numbers, mailboxes. Regular expressions are relatively inexpensive to some extent, so combining them with ai technology can improve efficiency and reduce time

2.2.3 Nearest neighbor classification

Sentence from training data will be transformed to word vectors which can be processed by the program. Word vectors are real numbers which words mapped to. These data have been manually labeled into corresponding intent classifications, such as greet, query about weather and so on. Sentences from test data will also be transformed to word vectors after the same processing. Then calculate the distance between the word vectors of test data and training data. The intent of word vector of training data which is closest to the word vector of test data will be considered as the most likely intent of the test data. This method is similar to KNN, both of them calculate the distance in the same space to obtain the intent.

2.2.3 support vector machine

SVM introduces the support vector machine method of machine learning based on the above-mentioned nearest neighbor classification method to recognize intent. Support vector machine shows a lot of advantages in dealing with small-sample and nonlinear data, and can be applied to other machine learning problems like function fitting. Its main idea is to map a linearly inseparable set of sample data to a highdimensional space to make it linearly separable, and the kernel function can process data when the data representation of the high-dimensional space is unknown.

The machine learning algorithm which Rasa NLU uses to classify intents is the SVM with GridSearchCV.

2.2 Named Entity Recognition

The machine learning algorithm which Rasa NLU uses to classify intents is the SVM with GridSearchCV.

For extracting unknown entities, keyword matching is not suitable, because it is impossible to pre-define every entity. his part of work can also be achieved by regular expression. It is simple, but it is difficult o define a suitable regular expression. Rasa NLU and spaCy have provided methods to extract entities, but other recognize methods such as spelling, context and after specific words also need to be used. Because similar identifies may play different roles in sentences. For example: Tell me about the historical open data of Apple from 2018-11-1 to 2018-11-10. In this sentence, "2018-11-1" and "2018-11-10" are both date, but one is start time, and another is end time. Synonyms should also be considered in. For example, "Apple" and "AAPL" are the same stock.

2.3 Multiple round multiple query technique

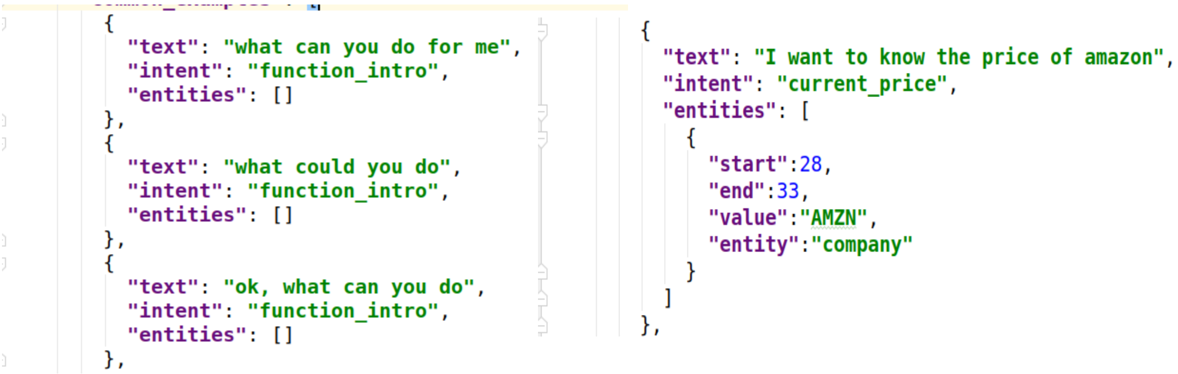
The state machine can control the state information of the robot, so as to realize different structures, such as progressive layer, loop or undetermined, etc., making multiple rounds of multiple queries possible, which also makes the robot more humanized and increases the complexity.

3 IMPLEMENTATION

3.1 Training Data and Model

The Rasa NLU is used as the main training tool. The training set need to be established first. The data type used in this project is Json, the format is:

The approach to achieve extracting intent and entities is machine learning. Rasa-NLU is selected as the main tools. First, install Rasa-NLU packages in the environment and import it in the project. Rasa-NLU accepts different kinds of training data. The data type used in this project is Json which is more readable and has a clear structure. The standard format of JSON type data is shown below:



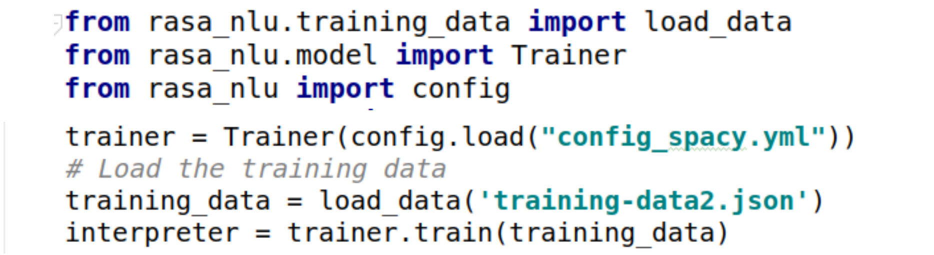
```
{
  "text": "what can you do for me",
  "intent": "function_intro",
  "entities": []
},
{
  "text": "what could you do",
  "intent": "function_intro",
  "entities": []
},
{
  "text": "ok, what can you do",
  "intent": "function_intro",
  "entities": []
},
```

```
{
  "text": "I want to know the price of amazon",
  "intent": "current_price",
  "entities": [
    {
      "start":28,
      "end":33,
      "value":"AMZN",
      "entity":"company"
    }
  ]
},
```

In theory, the training data set should as large as possible. However, only dozens of sentences are defined for an intent in this project due to the limited time. With the help of Rasa-NLU, an interpreter can be trained in only serval lines code. A well-trained interpreter can extract the intent and entities in different natural language sentences.

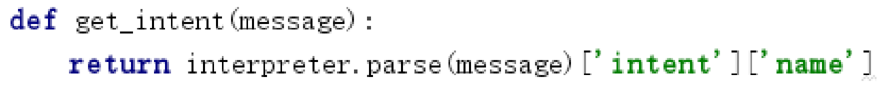
3.2 Intent Extraction and entities recognition

After training an interpreter, interpreter.parse(message)can be used to extract intent and entities of a sentence. The code is shown below:



```
from rasa_nlu.training_data import load_data
from rasa_nlu.model import Trainer
from rasa_nlu import config

trainer = Trainer(config.load("config_spacy.yml"))
# Load the training data
training_data = load_data('training-data2.json')
interpreter = trainer.train(training_data)
```

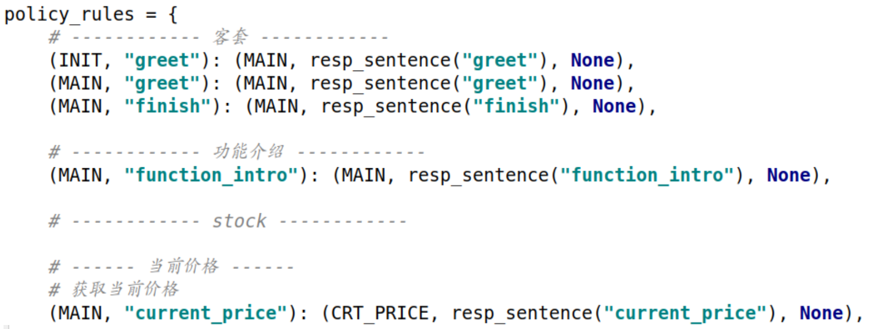


```
def get_intent(message):
    return interpreter.parse(message)['intent']['name']
```

3.3 Policy Rules and Respond

The chatbot will give a corresponding response to each sentence from the other side. If the chatbot receive a vague query and cannot get enough information, it will ask for additional information. Then the next step will be do the query based on the information from more than one rounds or quit this query.

The first value is the next system state; the second value is the reply should be sent to users. The code is as below shows:

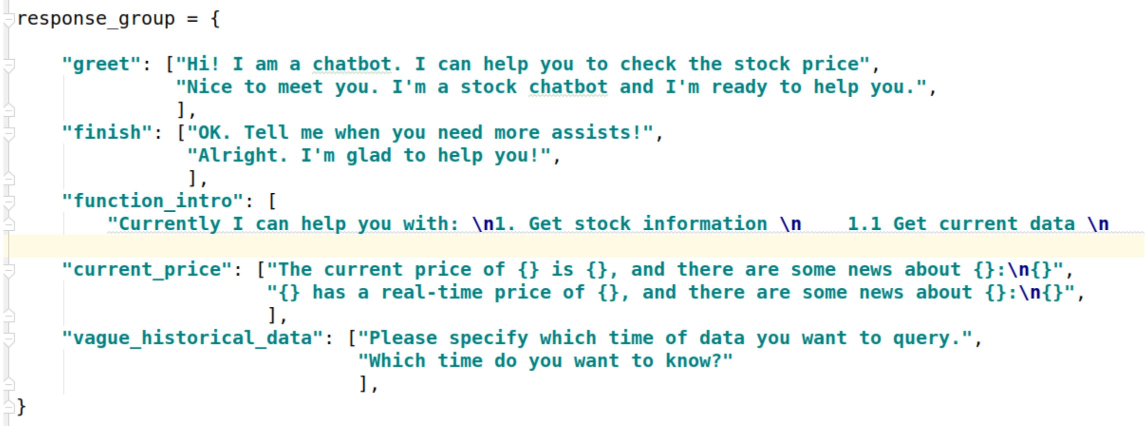


```
policy_rules = {
    # ------------ 客套 ------------
    (INIT, "greet"): (MAIN, resp_sentence("greet"), None),
    (MAIN, "greet"): (MAIN, resp_sentence("greet"), None),
    (MAIN, "finish"): (MAIN, resp_sentence("finish"), None),

    # ------------ 功能介绍 ------------
    (MAIN, "function_intro"): (MAIN, resp_sentence("function_intro"), None),

    # ------------ stock ------------

    # ------ 当前价格 ------
    # 获取当前价格
    (MAIN, "current_price"): (CRT_PRICE, resp_sentence("current_price"), None),
```



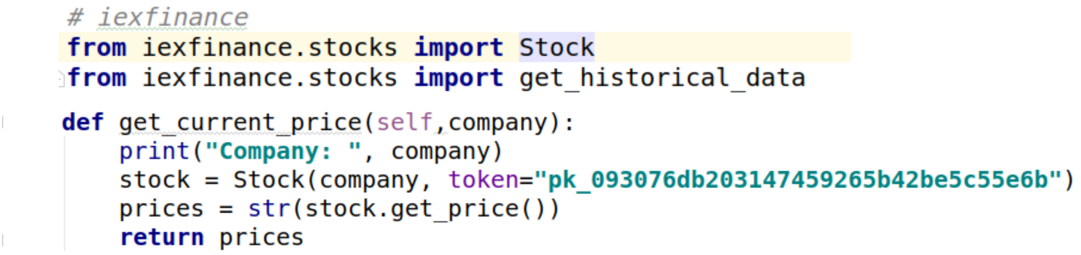
```
response_group = {

    "greet": ["Hi! I am a chatbot. I can help you to check the stock price",
              "Nice to meet you. I'm a stock chatbot and I'm ready to help you.",
              ],
    "finish": ["OK. Tell me when you need more assists!",
               "Alright. I'm glad to help you!",
               ],
    "function_intro": [
        "Currently I can help you with: \n1. Get stock information \n    1.1 Get current data \n

    "current_price": ["The current price of {} is {}, and there are some news about {}:\n{}",
                      "{} has a real-time price of {}, and there are some news about {}:\n{}",
                      ],
    "vague_historical_data": ["Please specify which time of data you want to query.",
                              "Which time do you want to know?"
                              ],
}
```

3.4 toolkit to obtain data for Stocks

The chatbot created in this project aims to help the user check the information about the stock. This function is achieved by using iexfinance API. This API can return information about stocks. It is simple to install it into the environment and import it into the project(Documentation: https://addisonlynch.github.io/iexfinance/stable/stocks.html). This project obtain the current price, historical price, recent news and TTM of a specific stock from this package. The code for get current stock price is as below shows:

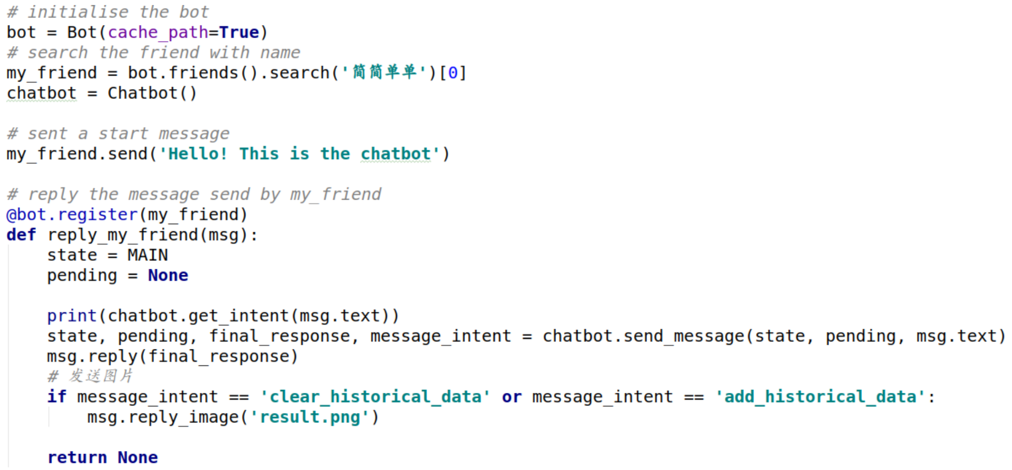


```python
# iexfinance
from iexfinance.stocks import Stock
from iexfinance.stocks import get_historical_data

def get_current_price(self,company):
    print("Company: ", company)
    stock = Stock(company, token="pk_093076db203147459265b42be5c55e6b")
    prices = str(stock.get_price())
    return prices
```

3.5 Wechat API

The final stage is to connect the program to WeChat which is known as the most popular online chat application in China. The tools used in this project is WXPY which is a powerful tool to process messages on WeChat. As usual, first, install the packages of WXPY to the Python environment and import to the project. Package wxpy need to be installed and import into the project first. A QR code will be generated after creating a bot. Then register this bot and it will be able to receive messages from specific friends and use the program to respond



```python
# initialise the bot
bot = Bot(cache_path=True)
# search the friend with name
my_friend = bot.friends().search('簡簡単単')[0]
chatbot = Chatbot()

# sent a start message
my_friend.send('Hello! This is the chatbot')

# reply the message send by my_friend
@bot.register(my_friend)
def reply_my_friend(msg):
    state = MAIN
    pending = None

    print(chatbot.get_intent(msg.text))
    state, pending, final_response, message_intent = chatbot.send_message(state, pending, msg.text)
    msg.reply(final_response)
    # 发送图片
    if message_intent == 'clear_historical_data' or message_intent == 'add_historical_data':
        msg.reply_image('result.png')

    return None
```

them.

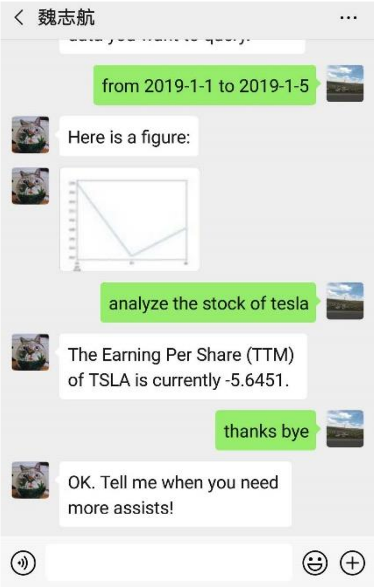
First, initialize a Bot() which can provide a QR code for the user to log in their WeChat account. In order to keep receiving the message from a specific friend or a group of them, registering the bot is a good choice. Then define a method called reply_my_friend(msg) to accept messages and send the answers to users.

4 RESULT AND DISCUSSION

An example dialogue between user and chatbot is shown below:

This chabot can answer the queries about stocks, it can also recognize the intent of "greet" and "finish" and choose suitable sentence to respond. This chatbot completed all the basic functions. However, because of the lack of trainning data and the rough process of language, there is still many limitations for this chatbot. It cannot recognize the intent and entitise of some sentence and its responses are still a little weired.

There is a wild range for natural language analysis chatbot since it can understand different intent if related training data is defined. For example, it can help people manage their schedules. However, a large number of training data need to be defined to make the chatbot cover different areas, which is a hard work needs to be finished by a large group of engineers.



from 2019-1-1 to 2019-1-5

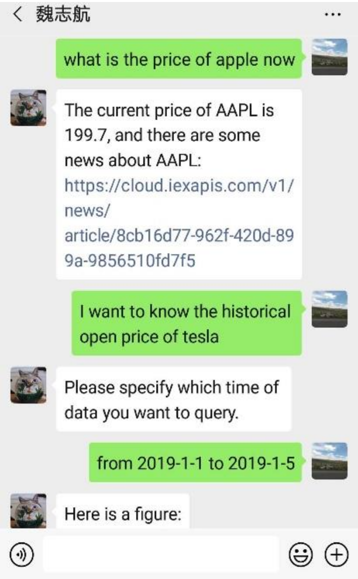
Here is a figure:

analyze the stock of tesla

The Earning Per Share (TTM) of TSLA is currently -5.6451.

thanks bye

OK. Tell me when you need more assists!



what is the price of apple now

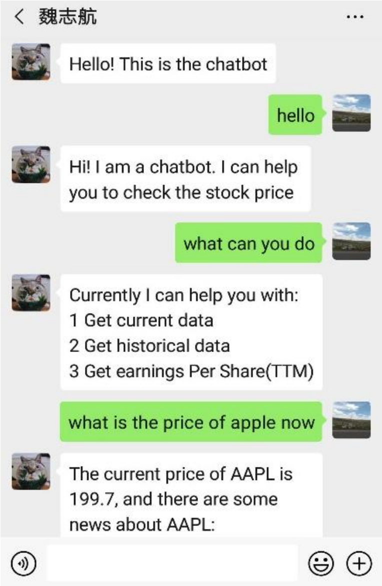
The current price of AAPL is 199.7, and there are some news about AAPL: https://cloud.iexapis.com/v1/news/article/8cb16d77-962f-420d-899a-9856510fd7f5

I want to know the historical open price of tesla

Please specify which time of data you want to query.

from 2019-1-1 to 2019-1-5

Here is a figure:



Hello! This is the chatbot

hello

Hi! I am a chatbot. I can help you to check the stock price

what can you do

Currently I can help you with:
1 Get current data
2 Get historical data
3 Get earnings Per Share(TTM)

what is the price of apple now

The current price of AAPL is 199.7, and there are some news about AAPL:

The effect of natural language chatbot in practical is pretty good, like Siri and Cortana, they can help users do a lot of works. On the contrary, some people complain about speech assistants are not smart enough. The reason for this is the training data set is not large enough. In fact, the performance of the chatbot is partly depended on the size of the data set. However, it is impossible to define a perfect data set covering all aspects. In order to improve the performance of the chatbot, on the one hand, refine the training data set, one the other hand, focus on a new machine learning algorithm which is harder to achieve.

5 CONCLUSION

The result shows that good chatbot can help users do numbers of works which improves their work efficiency. It can be applied in various area and have great potential. It also shows that it is not easy to build and smart chatbot. There are still some limitations on how to improve the performance of the chatbot. With the development of techniques and scientists' effects, natural language analysis chatbot will become real smart in the near future.

In this project, I learned a lot of new knowledge and successfully applied it to the chatbot. The course content includes from the initial regular expression to the final state machine multiple rounds of queries, I gradually understand the construction process of chatbot under the guidance of my mentor, Zhang Fan.

At the same time, the teacher also gives us a lot of freedom, let us acquire knowledge independently, which greatly improves our practical skills. If you do not know how to use the API, you can query its documentation. Facing difficulties, I get lots of help from my mentor. In the process of completing this project, I became more proficient in python and more familiar with spacy, rasa language frameworks.

Although the training model is not perfect, semantic recognition is sometimes inaccurate. But overall, I felt a great sense of accomplishment when I finished a chat bot that could be of practical use.