Football AI
Chat Robot on Wechat

Jiadong Lou
Advisor: Fan Zhang
2018/10/16

Table of Contents

1. Introduction ……. 2
2. System Overview ……..……..…….. 2
   - 2.1. WXPY ……....……..... 3
   - 2.2. Voice Translations ……...…….…….…….... 3
   - 2.3. Language Translation …….…...…….. 3
   - 2.4. RASA_NLU with Jieba and MITIE …….....……..…… 4
   - 2.5. Events API ……....…… 5
3. Simulation …….……....…… 5
4. Installation …….……...… 7
5. Conclusion ……..... 7
6. Acknowledgement …….…….…….. 7

1. Introduction

For those who love watching football(soccer) games, it's difficult to get live data about a mactch . You will eithier need to watch a live streaming on TV or use sports applications on your phone. However, Football AI allows people to get data of a match with a simple text message or voice message on Wechat platform. The system supports both Chinese and English searching and includes majority of football games around the world. This system can turn any individual Wechat account into a smart and intelligent Football AI robot.

2. System Overview

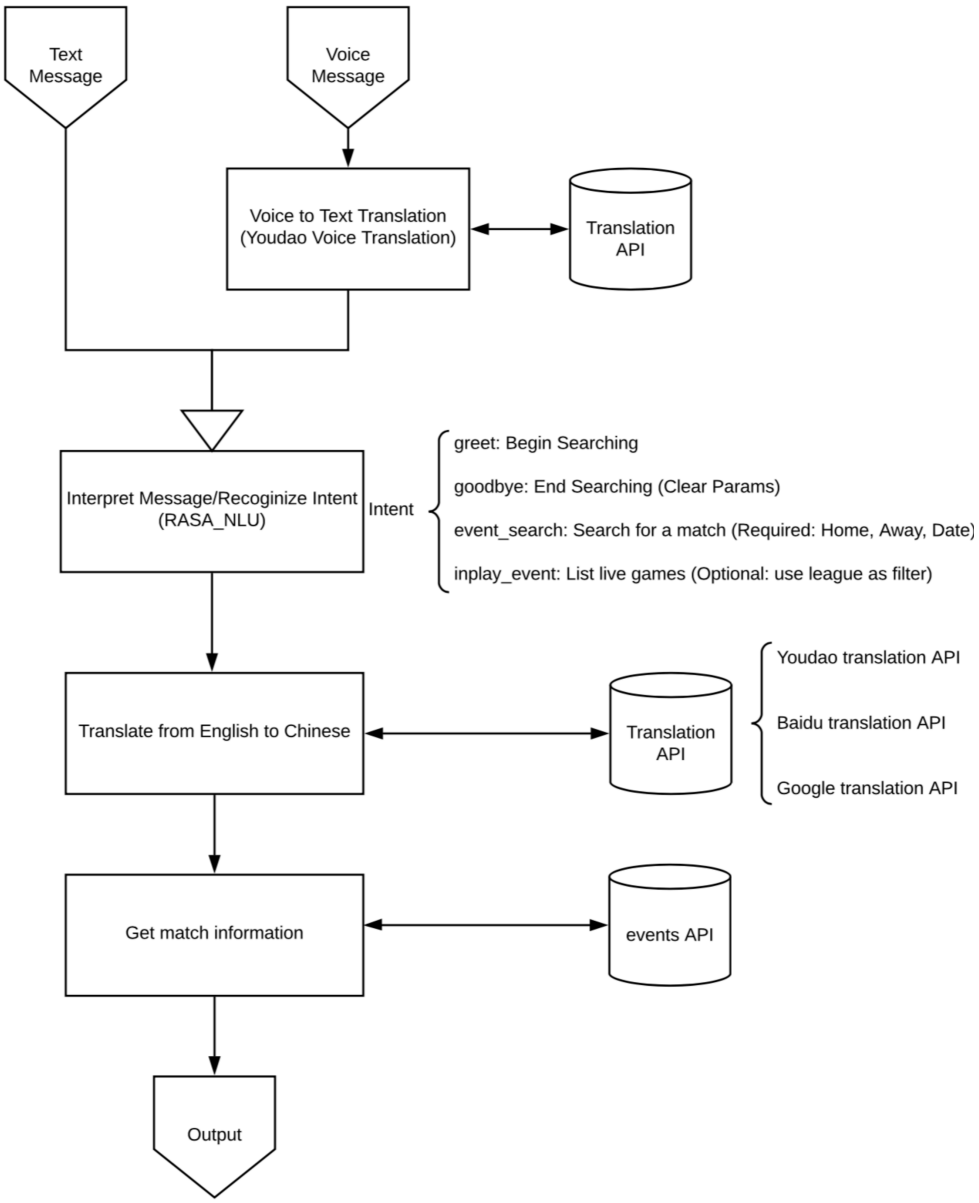

Figure 1. System Overview Diagram

2.1 WXPY

wxpy is a package that allows program to get message from Wechat and reply message accordingly. Once logged in, wxpy actively waits for any message. When a message is received by wxpy, the respond function will be called. When the respond function is returned, the wxpy will send the response back to the message sender.

```
Getting uuid of QR code.
Downloading QR code.
Please scan the QR code to log in.
Please press confirm on your phone.
Loading the contact, this may take a little while.
Login successfully as JAYDEN
```

Figure 2. Prompt for Wechat Log In using wxpy

2.2 Voice Translation

Youdao Translation ASR is used for translating Voice message into Text message. The voice message is saved as mp3 file. Then it's converted to wav file and passed to Youdao API. The translated message is extracted from a Json file.

```
def youdao_voice_translate(source, fromLang, sample_rate, nchannels):
    url = 'http://openapi.youdao.com/asrapi'
    salt = random.randint(1, 65536)
    q= source
    sign = appKey+q.decode('utf-8')+str(salt)+secretKey
    sign = sign.encode('utf-8')
    m1 = hashlib.md5()
    m1.update(sign)
    sign = m1.hexdigest()

    dict = {'english' : 'EN', 'chinese' : 'zh-CHS'}
    fromLang = dict[fromLang]

    data = {'langType': fromLang, 'q':q,'doctype': 'json', 'appKey': appKey,'salt':salt,
    'sign':sign, 'format': 'wav', 'rate': sample_rate, 'channel':nchannels, 'type':1}

    data = urllib.parse.urlencode(data).encode('utf-8')
    wy = urllib.request.urlopen(url, data)
    html = wy.read().decode('utf-8')
    result = json.loads(html)

    return result['result'][0]
```

Figure 3. Code Sample of Voice Translation

2.3 Language Translation

The program needs language translation between Chinese and English. It has the option to use Google Translation API, Youdao Translation API, or Baidu Translation API, depending on location of the server.

2.4 RASA_NLU with Jieba and MITIE

Rasa NLU is an open-source natural language processing tool for intent classification and entity extraction in chatbots. Using the Jieba tokenizer together with MITIE allows training a model in Chinese language. Therefore, the program will be able to extract intents and entities from message in Chinese. RASA_NLU provides accurate intent extraction with confidence level. The program uses confidence level to avoid unintentinal trigger or spam messages. Any message with confidence level lower than 0.45 will be ignored.

```
{
  "text": "查询今天凤凰城对拉斯维加斯的比赛",
  "intent": "event_search",
  "entities": [{
      "start": 2,
      "end": 4,
      "value": "今天",
      "entity": "date"
    },
    {
      "start": 4,
      "end": 7,
      "value": "凤凰城",
      "entity": "home"
    },
    {
      "start": 8,
      "end": 13,
      "value": "拉斯维加斯",
      "entity": "away"
    }]
},
```

Figure 4. Training Examples given to RASA_NLU

```
Training to recognize 5 labels: 'home', 'away', 'date', 'data', 'league'
Part I: train segmenter
words in dictionary: 200000
num features: 271
now do training
C:           20
epsilon:     0.01
num threads: 1
cache size:  5
max iterations: 2000
loss per missed segment:  3
```

Figure 4. Training Process in Terminal

The training process normally will take 20 to 30 minutes depending on the performance of the computer.

2.5 Events API

Bets API is a RESTful service for data on all sports. This program uses EVENTS API in order to search for data of a specific match or get a list of inplay matches. For event search, the user needs to provide the name of Home Team, Away Team, and Date of the match.

URL Parameters

Parameter	Required?	Description
sport_id	Yes	Reference
home	Yes	home team ID or name
away	Yes	away team ID or name
time	Yes	either UTC time epoch (Limited to 90 days) or day YYYYMMDD

Figure 5. URL Parameters for event_search

For in_play event, the program will request a list of inplay matches and then print out 15 of selected matches. If there are more than 15 inplay matches, the program will select most recent 15 matches from major leagues. If there are less than 15 inplay matches, the program will print all matches from major leagues and then fill up the rest with matches from non-major leagues. If there are no inplay matches, the program will remind user no inplay matches. User can also look for inplay matches from a specific league. In this case, the program will only return matches for that specific league.

```
def inplay_event():
    search_url = "https://api.betsapi.com/v1/events/inplay?&token={}"
    search_url = search_url.format(token)
    params = {
        'sport_id' : 1
    }
    return requests.get(search_url,params = params).json()
```

Figure 6. Code Sample for inplay_event

3. Simulation

Multiple tests and simulations have been performed. The results indicate that the program works as expected.

Figure 7. Simulation of event_search for scores

Figure 7 shows that the program is able to find the score of the requested match in both Chinese and English. The simulation in Chinese also involves with Voice translation.

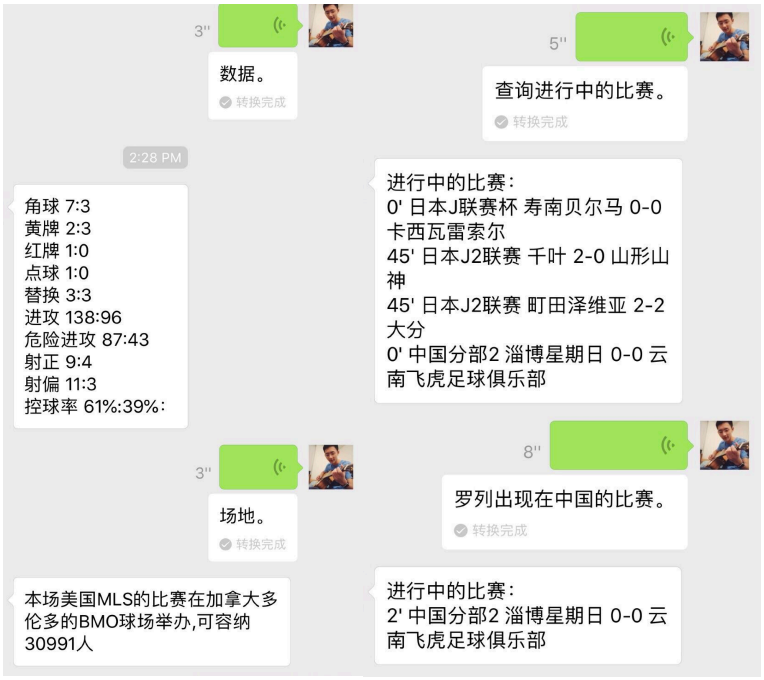

Figure 8. Simulation of search_event and inplay_event

Figure 8 shows that the program is able to find the data and location of the match. It can also list all inplay matches or only inplay matches from a country/league.

4. Installation

1. Clone code from Github

Github Link: https://github.com/jiadonglou/football_AI

2. Download and save dat file to ./data/

Download Link: https://pan.baidu.com/s/1NX-dZ5ONDEDYhZ0-PdW5Ng

Password: g7cw

3. Set up environment (using anaconda3)

Environment file is save at ./environment.yaml

```
conda env create -f environment.yaml
```

4. Run the program

```
python main.py
```

5. Conclusion

Presently, there is no easy way to get sports data on Wechat. With this program, people can access live scores or statistics of match while they are driving, exercising, or working. With the help of voice translation, people can talk to the program and get the information they needed. However, the program now only supports limited functionalities. In the future, the program can be developed to support more dynamic inputs, more diverse data inquries and a customized notification.

6. Acknowledgments

I would like to thank my advisor Fan Zhang for teaching me the knowledge about Natrual Language Processing and guiding me throughout this project. I have learned a lot about RASA_NLU, getting data from API, and programing in general.