ReportsCompilation

November 13, 2018

1 WeChat Report Compilation

A compilation of 8 reports.

1.1 Setup

API, environment and encryption packageg

- Use package ItChat instead of wxpy
- ItChat last commit - 9 months ago, 13378 star
- wxpy last commit - a year ago, 6866 star
- Use deveplopment environment IPython 3 (jupyter)
- web-based interactive development environment
- easy to demo and make changes
- Use end-to-end encryption in chats
- prevent interference from server side

1.2 Tests

in 4 steps, from small scale to lareg scale

0. Setup Wechat account API base on Wechat Protocol and package capture

- Obtain 10 WeChat Id and maintian them
- Test with friends (small scale)

1. Setup 1-to-1 chat

- Obtain Wechat id in .csv
- Add friends by Wechat id
- Set nickname to subject according to experiment design (anonymous)

2. Send test message

- Setup individualized test message
- Require participant to repond by sending the same message back
- Check for returned message

3. Set up end-to-end encryption

- Test with friends
- Test with participants

1.3 Issues

tested workable solution marked as checked

- [X] Handle issues and reports
- set up Help option in automatic reply to report mess-ups
- set up keywords to filter out junk information
- [ ] Need to scan QR code to login.

- install TKkk-iOSer/WeChatPlugin-MacOS plugin to auto-login
- [X] Prevent auto-logout and crashes
- keep login-ed, and keep a back-up cache.
- [ ] Censorship
- use end-to-end encryption, pyca/cryptography
- wechat-encrypt
- [X] Survey
- use wjx
- accessible and fast, email-login, WeChat-distributable, tracks user IP
- supports large sample size, logic loops and other survey features

1.4 Notes

for setting up server, data analysis, extending functions etc. later

- Server setup
  - link the WeChat server to the chatbot backend with HTTPS
  - HTTP GET and HTTP POST requests to transfer the messages between the chatbot and WeChat sever.
  - Python dependencies might be needed
    * Gunicor (Python web server),
    * falcon (web API framework for Python)
    * xmltodict (a library making XML feel like working with JSON)
  - Apply two kinds of dynos of Heroku to handle the synchronous and asynchronous replies
    * web dynos to handle the frontend work
    * worker dynos to process the queued jobs in background.
- Data analysis
  - Matplotlib for python/matlab is an excellent library handling data;
  - Chart.js for web/html5 is good for presenting statistics.
- Chat AI Chatterbox
  - needs training data
- Other WeChat chatbot for reference
  - PHP based vbot
  - Perl based Mojo-Weixin

```
In [ ]: # install wxpy Python package as a shell command
        !pip install -U wxpy
```

```
In [2]: #FIXME
        # demo code using wxpy
        from __future__ import unicode_literals
        from threading import Timer
        from wxpy import *
        import requests

        bot = None
        def get_news():
            url = "http://open.iciba.com/dsapi/"
            r = requests.get(url)
            print(r.json())
            contents = r.json()['content']
            translation = r.json()['translation']
            return contents,translation

        def login_wechat():
            global bot
            bot = Bot()
            # bot = Bot(console_qr=2,cache_path="botoo.pkl")#For linux

        def send_news():
            if bot == None:
                login_wechat()
            try:
                my_friend = bot.friends().search(u'liz')[0]
                my_friend.send(get_news()[1][5:])
                my_friend.send(u"hello world")
                t = Timer(60, send_news) #1 min
                t.start()
            except:
                print(u"failure!")
            if __name__ == "__main__":
                send_news()
                print(get_news()[0])
```

```
In [ ]: # demo code using wxpy
        from wxpy import *
        bot = Bot()

        # search for friend named 'Friend' and send message
        friend = bot.friends().search('Friend')[0]

        # send a message
        friend.send(u"Hello, World!")
```

```
# automate reply
@bot.register(friend)
def reply_friend(msg):
    msg.reply(u'Good morning, good evening and good nite.')

# keep login-ed
embed()
```