

OSSP 3 차 진행사항 보고서



2020.11.22

동국대학교 Team USS

목차

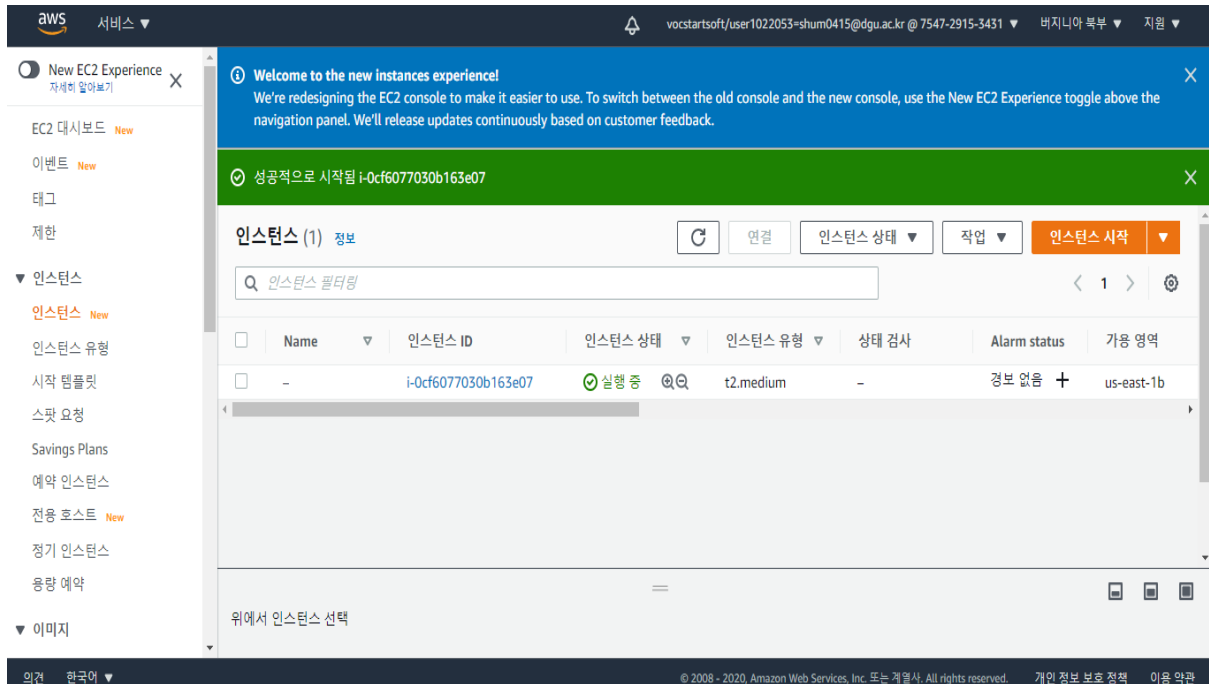
- 이전 진행사항
 - 챗봇 모델 개발
 - 카카오톡 대화기록 데이터 전처리
 - 카카오톡 API 연동
 - 어플 개발
 - 피드백
- 현재 진행사항
 - aws 를 이용한 가상환경 구축
 - 카카오톡 연동 2 가지 방법
- 향후 일정
 - 중간발표
 - 이후

□ 이전 진행사항

- 챗봇 모델 개발
 - 코드 에러 부분을 수정하여 챗봇 소스 코드 실행을 성공.
 - 아직 완벽하게 예측 값을 나타내지 못하여 추가적인 모델 분석 및 수정이 필요.
- 카카오톡 대화기록 데이터 전처리
 - 대화기록 .txt 파일 형식을 .csv 파일 형식으로 바꿔주는 코드를 개발.
- 카카오톡 API 연동
 - 카카오톡 API 코드를 확인하고 필요에 맞게 수정.
 - 친구목록 추출하기
 - 메시지 보내기
 - 카카오톡으로 보내기
- 어플 개발
 - 안드로이드 스튜디오를 이용.
 - 아직 기초적인 단계로 추가적인 학습을 통해 디자인과 코드 개발 필요.
- 피드백
 - 챗봇에 대해 너무 거창한 느낌
 - 챗봇 보다는 존댓말, 답장 시간 예측 등을 이용한 커스터마이징을 하여 자동응답기를 구현하는 것이 좋을 것 같음
 - 방향성을 확실히 정할 필요가 있음
 - 코드에 숫자가 들어 있으면 감점

□ 현재 진행상황

- aws 를 이용한 가상환경 구축
 - 자동응답기 파일을 구동 시키기 위한 가상환경 필요
 - Aws ubuntu 를 이용하여 구축
 - Xdrp 를 이용하여 gui 로 ubuntu 이용
 - ubuntu 내에 아나콘다 및 파이썬 설치



- 카카오톡 연동 2 가지 방법
 - 카카오톡 api 를 이용한 방법 (기존 방법)
 - html 파일을 이용한 웹 서버 개발

```

1 <!DOCTYPE html>
2 <html lang="ko">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>API 연습</title>
8 </head>
9
10 <body>
11   <h1>카카오톡 자동응답기 웹 서버</h1>
12   <script src="https://code.jquery.com/jquery-3.5.1.js"
13     integrity="sha256-QWo7LDvxbWT2tbbQ97B53yJnYU3WhH/C8ycbRAKjPDc=" crossorigin="anonymous"></script>
14
15   <script>
16     $.ajax({
17       method: "GET",
18       url: "https://kapi.kakao.com/v1/api/talk/friends?limit=2",
19       data: { friend_order: "nickname" },
20       Headers: {Authorization: "Bearer {e3b5ccea84bb44772a6b12dc7afaf712}"},
21     })
22     .done(function (msg) {
23       console.log(msg);
24     });
25   </script>
26
27 </body>
28
29 </html>

```

카카오톡 자동응답기 웹 서버

- 카카오톡 메시지 api 를 이용한 메시지 보내기

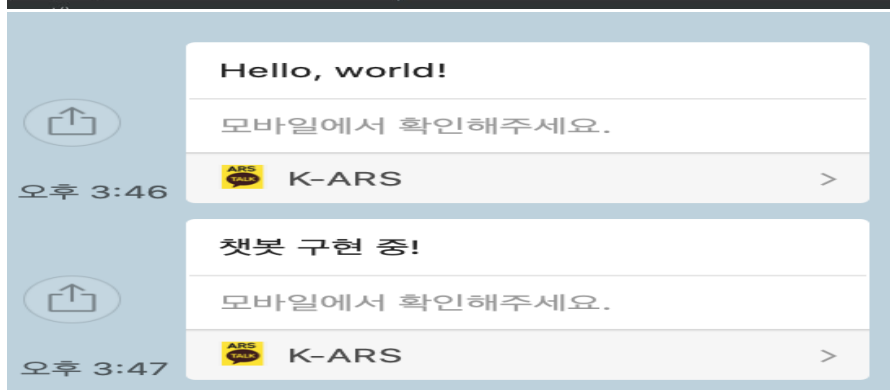
```

def message():
    url = "https://kapi.kakao.com/v1/api/talk/friends/message/default/send"

    # 사용자 토큰
    headers = {
        "Authorization": "Bearer " + "gjjyGfz1B9Awxf-KDcX8wuYkbb-jzn008dGmZMgorDSAAAAF1xX87kg"
    }

    data = {
        "receiver_uuids": ["16임순호"],
        "template_object": json.dumps({
            "object_type": "text",
            "text": "챗봇 구현 !",
            "link": {
                "web_url": "www.naver.com"
            }
        })
    }

```

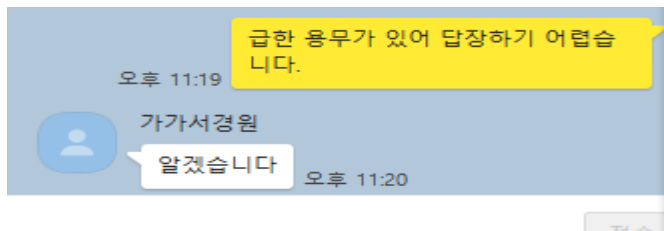


- 파이썬으로 pc 카카오톡을 제어해서 메시지 보내기
 - (동기) 애플리케이션을 이용한 자동 응답기는 커스터 마이징에 취약하고 접근 권한 때문에 활용에 부정적
 - 웹 서버나 pc 버전 카카오톡을 이용하여 자동 응답기를 구현하기 위한 목적으로 방법을 탐구
 - 오픈소스 <https://gitlab.com/Whackur/kakaoautomsgsender/-/tree/master/> 에서 코드를 참고하여 코드를 수정하고 테스트
- 아래 실행 방법 및 코드
 - 메인 파일을 실행시키면 메시지를 보낼 카카오톡 친구 이름 받음
 - 이미지에선 예시로 “가가”로 했지만 실제로는 “가족”, “어머니”, “아버지”로 설정
 - 메시지를 보낼 카카오톡 친구 수 (범위) 설정
 - 전송할 메시지 내용 입력 받음 (코드를 수정하여 자동응답기 메시지 내용을 사전에 설정하고 대화 상대에 따른 커스터 마이징)
 - Pc 가 마우스 및 키보드를 제어하여 pc 카톡에서 친구 선택 후 대화창에 사용자에게 입력 받은 메시지를 복사 후 전송

```
(base) C:\Users\User>code
(base) C:\Users\User>cd desktop
(base) C:\Users\User\Desktop>cd kakaoautomsgsender
(base) C:\Users\User\Desktop\kakaoautomsgsender>python my_main.py
Monitor size : Size(width=1366, height=768)
Point(x=826, y=181)
필터링할 친구 이름, 없으면 enter, ex) 학생 직장 99 : 가가
필터링한 친구 기준 시작지점(ex, 필터링된 친구 시작지점) : 1
반복할 횟수(ex, 필터링 검색된 친구 수) : 2
전송할 메시지 : 급한 용무가 있어 답장하기 어렵습니다.

(base) C:\Users\User\Desktop\kakaoautomsgsender>python my_main.py
Monitor size : Size(width=1366, height=768)
Point(x=617, y=231)
필터링할 친구 이름, 없으면 enter, ex) 학생 직장 99 : 가가
필터링한 친구 기준 시작지점(ex, 필터링된 친구 시작지점) : 1
반복할 횟수(ex, 필터링 검색된 친구 수) : 2
전송할 메시지 : 급한 용무가 있어 답장하기 어렵습니다.

메세지 전송 시작!
Repeat Number : 1 // Time wait : 4.4539395974673495
Repeat Number : 2 // Time wait : 4.587773363116242
프로그램이 종료되었습니다.
```



- 소스코드 이미지

```

my_main.py X
my_main.py
1  import sys
2  import pyautogui
3  import time
4  import pyperclip
5  import os
6  import random
7
8
9  def send_msg(my_msg, repeat_number):
10     for i in range(int(repeat_number)):
11         time_wait = random.uniform(3, 5) # 카카오톡 측에서 자동화를 방지할 수도 있기 때문에 랜덤하게 3에서 5초 사이 간격을 줌
12         print('Repeat Number : ', i + 1, end='') # 반복 횟수 0에서 시작하기 때문에 1을 더해줌
13         print(' // Time wait : ', time_wait)
14         time.sleep(time_wait)
15         pyautogui.keyDown('enter')
16         pyperclip.copy(my_msg)
17         pyautogui.hotkey('ctrl', 'v')
18         pyautogui.keyDown('enter')
19         pyautogui.keyDown('esc')
20         pyautogui.keyDown('down')
21
22
23  def filter_friend(filter_keyword, init_number):
24     # 사람 아이콘 클릭
25     try:
26         click_img(img_path + 'person_icon.png')
27     except Exception as e:
28         click_img(img_path + 'person_icon2.png')
29     except Exception as e:
30         print('e ', e)
31     except Exception as e:
32         print('e ', e)
33     # x 버튼이 존재한다면 클릭하여 대화 삭제
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71  def initialize():
72     print('Monitor size : ', end='')
73     print(pyautogui.size()) # 화면 사이즈 출력
74     print(pyautogui.position()) # 마우스 포인트 x, y로 좌표 받아옴
75     filter_keyword = input("필터링할 친구 이름. 없으면 enter. ex) 학생 직장 99 : ")
76     init_number = input("필터링할 친구 기준 시작지점(ex. 필터링된 친구 시작지점) : ")
77     repeat_number = input("반복할 횟수(ex. 필터링 검색된 친구 수) : ")
78     my_msg = input("전송할 메시지 : ")
79     print('=====')
80     print('메세지 전송 시작!')
81     print('=====')
82     return (filter_keyword, init_number, repeat_number, my_msg)
83
84
85  # config
86  img_path = os.path.dirname(os.path.realpath(__file__)) + '/img/'
87  conf = 0.90 # opencv의 이미지 정확도 최대 1
88  pyautogui.PAUSE = 0.5 # 전역 딜레이로 키보드와 마우스를 제어 하는데 매 행동 사이에 0.5초 간격을 줘서 행동을 구분할 수 있게 최적화를
89
90  if __name__ == "__main__":
91     (filter_keyword, init_number, repeat_number, my_msg) = initialize()
92
93     filter_friend(filter_keyword, init_number)
94     send_msg(my_msg, repeat_number)
95     bye_msg()
96

```

□ 향후일정

● 중간발표

- Pc 카톡을 제어하여 카카오톡 메시지를 보내는 것에 대한 활용방안 (추가 또는 제거)
- 카카오톡 API 를 이용하여 자동응답기 웹 서버 가동 및 메시지 보내기
- 카카오톡 API 를 통해 친구 정보를 받고 존댓말 레이블링 같은 커스터마이징 구현

● 이후

- 친구 정보를 활용한 커스터 마이징 기능 추가
- 챗봇 모델의 활용 가능성