

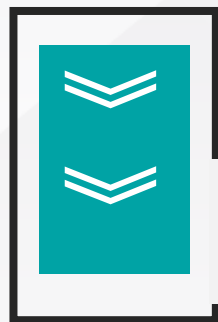
Project/JoyFarmer

項目進度階段性報告

野村投信二/第二組 20210411



台大電信研究所二年級 徐瑋辰
台大經濟系四年級 蕭羽平
東吳巨資三年級 韓昨非
東吳巨資二年級 楊品思
東吳巨資二年級 蔡明杰



目錄

CONTENTS

1

分工狀況

2

開會成果與檢討

3

與業師的討論原因和回饋成果

4

階段性成果

5

目前遇到的困難

01

分工狀況





第一階段分工

QASet製作

蕭羽平、楊品思、蔡明杰

初始架構設計和驗證

徐瑋辰、韓昨非

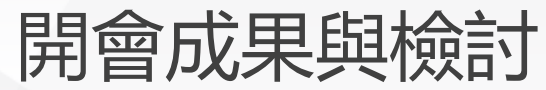
02

開會成果與檢討

03

與業師討論的原因





The calendar displays the following events:

- 20210321-pre-initiate** (2021-03-21)
- QASet-BERT-Faiss-API** (2021-03-21)
- 20210324-initial-meet** (2021-03-24)
- 野村ppt中的問答** (2021-03-24)
- 3/31 討論內容** (2021-03-31)
- 4/1 討論內容** (2021-04-01)
- 4/7 討論** (2021-04-07)
- 問題集** (2021-04-07)
- 4/9與mentor的討論** (2021-04-09)
- 4/9討論—問題集確認與進度規劃** (2021-04-09)
- 第一階段-PartI(情境式劇本確定)** (2021-04-09)
- 第一階段-PartI(機器人架構設計)** (2021-04-09)
- 4/14討論** (2022-04-14)
- 4/19東吳...** (2022-04-19)
- 4/25討論...** (2022-04-25)
- 0428討...** (2022-04-28)
- 0502討...** (2022-05-02)



開會成果與檢討/與業師討論的原因

| | | |
|------|------------|--|
| 0321 | 組員online | 理解題目概念、決定何時與業師offline討論 (Initiate) |
| 0324 | 與業師offline | 縮限範圍，確定是做靈活型問卷機器人，但是並沒有確定出題的細節 |
| 0331 | 組員offline | 取得了QASet的填充題和機器人原型，但是不知道如何讓機器人具備比現有機器人更好的體驗 |
| 0401 | 與業師online | 進一步瞭解了目標使用者的前置問卷填答狀況，明確了客戶畫像的量化指標應該往哪個方向發展，但是問的問題太少，沒有當場理解業師的回答並提出新的問題 |
| 0407 | 組員online | 明確兩點：QASet結構需要調整，問題涉及的基本原理和背景需要更加明確，需要和業師討論我們現有的理解是否正確以及詢問專案架構是否合理 |
| 0409 | 與業師online | 得知機器人的問卷盡量不要與金管會問卷重合，解決了出題的概念問題、理論問題，確定了專案步驟和機器人架構 |



開會成果與檢討/與業師討論的原因

| | | |
|------|-----------|---|
| 0414 | 組員online | 測試Telegram，但是發現QASet需要調整 |
| 0425 | 組員offline | 需要對QASet調整題目數量，並且根據Telegram的特點更改問答方式 |
| 0428 | 組員online | 討論風險計算公式以及如何處理開放式問答，攥寫測試回答 |
| 0502 | 組員online | 對題目的計分公式進行討論，發現個別題目需要調整提問方式或者調整分數，不是所有題目都應該岑與計算 |
| 0503 | 與業師online | 針對0502新發現的問題進行提問和確認 |

04

階段性成果





檔案 編輯 查看 插入 格式 資料 工具 說明 上次編輯是在數秒前





階段性成果——機器人局部驗證程式碼

```
print(rule)
jump_to = rule[redacted].iloc[0]['jump']
print(redacted.index.to_list()[0])
print(index)
distance = redacted.index.to_list()[0] - index
redacted
except:
    jump_to = df.iloc[index]['jump']
    distance = redacted.index.to_list()[0] - index
else:
    redacted
```

年齡

21

性別

male

情感狀況

solo

通勤

bus

外食or家裡

外食

answer

0 外食

1 家裡

5

4

餐館or夜市or超商or公司

餐館

覺得如何

好



阶段性成果——機器人局部驗證程式碼

Google 127.0.0.1 / MariaDB / apitest /

127.0.0.1/phpmyadmin/sql.php?db=apitest&table=answers&pos=0

phpMyAdmin

服务器: MariaDB:3306 » 数据库: apitest » 表: answers

正在显示第 0 - 8 行 (共 9 行, 查询花费 0.0001 秒.)

`SELECT * FROM `answers``

| | id | uid | answer |
|--------------------------|----|------------|---------------------|
| <input type="checkbox"/> | 1 | 758701283 | {'1.1.1': 'A', '1.2 |
| <input type="checkbox"/> | 2 | 1717753417 | {'1.1.1': 'a', '1.2 |
| <input type="checkbox"/> | 3 | 758701283 | {'1.1.1': 'A', '1.2 |
| <input type="checkbox"/> | 4 | 1717753417 | {'1.1.1': 'Any', '1 |
| <input type="checkbox"/> | 5 | 758701283 | {'1.1.1': 'A', '1.2 |
| <input type="checkbox"/> | 6 | 1717753417 | {'1.1.1': 'B', '1.2 |
| <input type="checkbox"/> | 7 | 758701283 | {'1.1.1': 'A', '1.2 |
| <input type="checkbox"/> | 8 | 758701283 | {'1.1.1': 'A', '1.2 |
| <input type="checkbox"/> | 9 | 758701283 | {'1.1.1': 'A', '1.2 |

api.php - apitest - Visual Studio

```
#TGBOT
Route::post('/tgbot',function (Request $request) {
    $data = ['uid' => $request->get('uid')];
    $record = App\Answer::create($data);
    return $record;
})->middleware('api');

Route::get('/answer/{id}',function ($request, $id) {
    $answer = App\Answer::find($id);
    return $answer['answer'];
})->middleware('api');
```

main* 0 0 0

下午 01:42 2021/5/3



階段性成果——機器人局部驗證程式碼

The screenshot displays a Windows development environment with three main components:

- File Explorer (Left):** Shows the directory structure of the project located at `D:\Programming\flask_api_gate`. The files listed include `_pycache_`, `TelegramFunction`, `app.py`, `app-https.py`, `chatbot.txt`, `fullchain.pem`, `judge.py`, `nlp.py`, `privkey.pem`, `processor.py`, `qa_tools.py`, `QASet.csv`, `readme.txt`, `session.py`, and `tg_logic_main.py`.
- Code Editor (Top Right):** Displays the `app-https.py` file in Visual Studio Code. The code is a Flask application that handles HTTP requests and interacts with a Telegram bot logic module.

```
1 from flask import Flask, request
2 from flask import jsonify
3 import json
4 import tg_logic_main
5
6 app = Flask(__name__)
7
8 @app.route('/hello')
9 def hello():
10     return jsonify(content="OK")
11
12 @app.route('/bot', methods=['POST'])
13 def bot():
14     data = json.loads(request.get_data(as_text=True))
15     res = tg_logic_main.flask_handler(data)
16     return str(res)
```
- Terminal (Bottom):** Shows the command prompt output for running the application. It indicates that the Flask app is being served on `https://0.0.0.0:443/` and displays a sample JSON payload received from a Telegram bot.

```
data:lab@DESKTOP-VVURH7P MINGW64 /d/Programming/flask_api_gate
$ activate env
data:lab@DESKTOP-VVURH7P MINGW64 /d/Programming/flask_api_gate
$ python app-https.py
* Serving Flask app "app-https" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on https://0.0.0.0:443/ (Press CTRL+C to quit)
[{"update_id": 437116389, "message": {"message_id": 246, "from": {"id": 758701283, "is_bot": False, "first_name": "昨非", "last_name": "韓", "username": "Hexplode", "language_code": "zh-hans"}, "chat": {"id": 758701283, "first_name": "昨非", "last_name": "韓", "username": "Hexplode", "type": "private"}, "date": 1620020684, "text": "A"}}]
91.108.6.93 - - [03/May/2021 13:44:47] "POST /bot HTTP/1.1" 200 -
```

參考資料

1. 專題整體架構

- a. <https://zaoldyeck.medium.com/%E5%AF%A6%E6%88%B0%E7%AF%87-%E6%89%93%E9%80%A0%E4%BA%BA%E6%80%A7%E5%8C%96-telegram-bot-ed9bb5b8a6d9>

2. telegram教學

- a. <https://herboratory.ai/telegram-chatbot-tutorial/>
- b. <https://hackmd.io/@truckski/HkgaMUc24?type=view>

3. telegram官方範例

- a. <https://github.com/python-telegram-bot/python-telegram-bot/tree/master/examples>

4. telegram問答集範例

- a. <https://pixnashpython.pixnet.net/blog/post/32391757-%E3%80%90telegram-api%E3%80%91python%E6%89%93%E9%80%A0telegame%E6%A9%9F%E5%99%A8%E4%BA%BA%E6%89%8B%E6%8A%8A%E6%89%8B%E6%95%99>

5. telegram雲端佈建範例

- a. <https://medium.com/%E8%AA%A4%E9%97%96%E6%95%B8%E6%93%9A%E5%8F%A2%E6%9E%97%E7%9A%84%E5%95%86%E7%AE%A1%E4%BA%BAzino/telegram%E8%81%8A%E5%A4%A9%E6%A9%9F%E5%99%A8%E4%BA%BA%E8%B6%85%E8%A9%B3%E7%B4%B0%E6%87%B6%E4%BA%BA%E5%8C%85-%E5%95%86%E7%AE%A1%E4%BA%BA%E9%83%BD%E7%9C%8B%E5%BE%97%E6%87%82-%E9%99%84python%E7%A8%8B%E5%BC%8F%E7%A2%BC-1ec81a91ce48>

05

目前遇到的困難





目前遇到的困難

如何設計計分公式

如何設計溫暖的反饋

對特定題目的開放性問
答設計精確的語義理解
模型



Project/JoyFarmer

謝謝觀看

野村投信二/第二組

