

# 專題實驗 進度報告

---

CYCU Instant Information Broker

10627110 王昱凱

10627211 林佑任

10627217 張哲睿

---

指導教授：吳宜鴻



---

# 目錄

## 1.簡介

## 2.專題介紹

## 3.實作方式

## 4.結論

## 5.未來發展

---

# 簡介

## CYCU Instant Information Broker

這是一個能夠收集有關中原大學的學生有關的資訊，並且加以分析。透過自然語言的意圖歸納，讓使用者能夠在日常的對答間，快速且精準的獲得想要的資訊。

## 專題說明

結合

*Line · Dialogflow · Heroku · MongoDB*

透過整合這些應用程式，讓我們能更有效的達成本專題的目標，使得這項成品能夠實際應用於生活之中。

我們將使用 Python 3.7 作為主要的程式語言，其中也將自己實作出爬蟲、文本分析、語意分析等等必要的功能。串接不同的介面，獲取所需的資訊來達成目標。

# 實作方式

## Line-Bot Server with Heroku and Dialogflow

透過 Line-Bot SDK with Python 來持續開發我們的後端 Server 並設計我們所需要的功能。目前我們將 Line-Bot Server 放置於 Heroku 平台持續運行以方便我們隨時更新測試。

```
# 監聽所有來自 /callback 的 Post Request
@app.route("/callback", methods=['POST'])
def callback():
    # get X-Line-Signature header value
    signature = request.headers['X-Line-Signature']
    # get request body as text
    body = request.get_data(as_text=True)
    app.logger.info("Request body: " + body)
    # handle webhook body
    try:
        handler.handle(body, signature)
    except InvalidSignatureError:
        abort(400)
    return 'OK'

# 處理訊息
@handler.add(MessageEvent, message=TextMessage) #每當收到LINE的訊息事件MessageEvent，且是一則文字訊息時，就執行下列程式碼。
def handle_message(event): #此函數接收LINE傳過來的資訊並貼上"event"標籤。

    # event長這樣是一個json物件
    # event = {"reply_token": "就是代表reply_token的一串亂碼",
    #         "type": "message",
    #         "timestamp": "1462629479859", 程式語言中標準的時間計算方式
    #         "source": {"type": "user",
    #                   "user_id": "就是代表user的一串亂碼"},
    #         "message": {"id": "就是代表這次message的一串代碼",
    #                   "type": "text",
    #                   "text": "使用者傳來的文字信息內容"}}

    #選擇參數text將他丟給dialogflow去解析
    responseJson = parse_user_text(event.message.text)

    #TextSendMessage是要執行的動作，LINE還提供了其他包括：ImageSendMessage、VideoSendMessage、StickerSendMessage等等的許多許多動作
    #message 也是一個json物件 (或許跟event長很像)
    #把message的"text"這個項目改成此訊息經由dialogflow解析後的action
    message = TextSendMessage(text = responseJson["result"]["parameters"]["action"])

    line_bot_api.reply_message(event.reply_token, message)
    #LineBotApi物件的reply_message只能在回覆訊息，且提供兩個參數:reply_token只能使用一次用完即丟
    #當其他使用者傳送信息給你的 LINE 聊天機器人，會產生一個reply_token，
    #你的聊天機器人拿著這個reply_token回覆傳信息的使用者，回覆完畢，reply_token消失

import os
if __name__ == "__main__":
    port = int(os.environ.get('PORT', 5000))
    app.run(host='0.0.0.0', port=port)
```

我們自己撰寫 Function 去擷取食用者輸入並傳送到 Dialogflow 來對使用者輸入進行語意分析，得到回傳資訊，了解使用者需求且透過 Python code 來處理後續的事項來達到我們想要完成的目。

```
def parse_user_text(text): #傳訊息給dialogflow並得到解析後的答案

    request = ai.text_request()
    request.query = text #欲查詢的字串
    response = request.getresponse().read().decode('utf-8') #解讀API回傳的JSON檔案(用UTF8解碼)
    responseJson = json.loads(response) #將json檔案轉為字典物件
    #字典結構: {'key': 值}
    #印出dialogflow分析我傳過去的字串的結果的其中我所想要的項目和其內容
    #print(responseJson["result"]["parameters"]["action"])
    #print(type(responseJson["result"]["parameters"]["action"]))
    return responseJson
```

# MongoDB

## 與 database 互動的窗口(main.py)

```
from pymongo import MongoClient
# from bson.objectid import ObjectId #這東西再透過ObjectId去尋找的時候會用到

# connection
def runMongo(data):
    client = MongoClient("mongodb+srv://johnson7543:deNBxJ@cluster0-84ii5.mongodb.net/Test?retryWrites=true&w=majority")
    db = client.Test
    collection = db.test2

    # test if connection success
    collection.stats # 如果沒有error, 你就連線成功了。
    print (data)
    # mydict = { "name": "YuKai Wang", "Email": "johnson7543@cycu.org.tw", "brith": "1998/09/21" }

    for temp in data :
        mydict ={"title": temp["title"], "href": temp["href"], "tweet": temp["tweet"], "author": temp["author"]}
        collection.insert_one(mydict)
```

存入的 data 暫時先使用自己寫的 PTT 爬蟲程式  
結果如下

```
QUERY RESULTS 1-20 OF MANY

{
  "_id": ObjectId("5ee1d7058a7f9a80b4244aca"),
  "title": "[閒聊] 彈射世界x公主連結連動",
  "href": "https://www.ptt.cc/bbs/C_Chat/M.1591858597.A.05F.html",
  "tweet": 5,
  "author": "s22499927"
}

{
  "_id": ObjectId("5ee1d7088a7f9a80b4244acb"),
  "title": "[討論] 七大罪到最後班和金雞比較強?",
  "href": "https://www.ptt.cc/bbs/C_Chat/M.1591858666.A.E93.html",
  "tweet": 0,
  "author": "dro001"
}

{
  "_id": ObjectId("5ee1d7088a7f9a80b4244acc"),
  "title": " ",
  "href": " ",
  "tweet": 0,
  "author": " "
}
```

## 目前成果

1. 能夠將資料透過程式存入 database，方便後續使用。  
目前能夠針對使用者所輸入的內容傳送到 Dialogflow 進行語意分析了解其意圖，並且回傳所分析出來的意圖訊息(下周有考試嗎?能夠分析出來它的意圖是要問考試)。



## 結論

我們目前的 Line 聊天機器人有一個基本的雛形，透過 Dialogflow 能夠簡單的瞭解到使用者的意圖。並且回傳其意圖的訊息給。我們也能透過爬蟲爬取我們所需的資料。

## 未來發展

未來我們要討論與選擇提供那些資料內容與結果資料如何呈現在 Line 的使用者介面上以方便使用者查看，在短時間內我們預計達到的目標如下圖所示，希望能夠準確判斷使用者語句的意圖與意思，並透過簡單明瞭的問答來做為篩選資料時的多項條件，並且將最後程式搜尋的結果回傳給使用者，以達到減少上網查詢資料所耗費的大量時間的目標。

