訊息理解與 Web 智慧

[作業一] Web crawler and Open source IR system

資工3B

108502584 陳顥升

cliffxzx@gmail.com

Abstract

使用 Python 的 scrapy 套件對巴哈姆特「動畫瘋」進行彈幕的蒐集，最後講蒐集到的資訊存入 elasticsearch 中，並且對彈幕分析。

Introduction

1. Web Scraping

   我使用 python 中的 scrapy 套件爬取資料，並且將處理完的資料用 elasticsearch 這個套件直接存到 elasticsearch，而無需人工匯入匯出，最後爬到了 3 萬多筆的彈幕。

   1.1 工具選擇

   在爬蟲工具的選擇上，我選擇使用 Scrpy 這個 python 套件，不用更常用的 BeautifySoup & (request | selenium) 原因是已下幾點

   a. 較 BeautifySoup 而言更具維護性，之後要擴增或修改都比較容易

   b. 不需考量過多設計，本身就相當有結構

   因為巴哈姆特「動畫瘋」的彈幕不需要 selenium 就可以爬取，所以選擇更有效率的方式。

2. elasticsearch 架設

   我使用 docker 將 elasticsearch 和 kibana 很方便的架起來，一開始遇到不同版本安全性的問題，最後找到方法解決了。

Method

1. 專案架構

   整個專案有 5 個檔案 AniGamerCrawler.py, AniGamerPipeline.py, DanmuItem.py, elk.py, main.py 功能分別如下

   a. AniGamerCrawler.py

      爬蟲的主要程式碼，有 3 大步驟，首先爬蟲會先去 “所有動畫”的頁面，抓取最大 page 數，並且開始跑所有頁數，將每一頁的動畫序號抓出來，下一步是進入每個動畫的頁面，將所有的動畫集數序號列出，接著已蒐集到的動畫集數序號取回彈幕，通通做完後會丟到 AniGamerPipeline.py。

   b. AniGamerPipeline.py

      這步驟是將以處理好的資料，創建一筆 index 存在 elasticsearch 。

   c. DanmuItem.py

      這是 scrapy 要求丟進 AniGamerPipeline.py 前需儲存的格式。

   d. elk.py

      這是存 elasticsearch new 出來 instance 讓其他文件共享。

   e. main.py

      主程式入口

2. 操作流程

3. 資料結構

```
'color': {
    'type': 'text',
    "fielddata": True
},
'position': { 'type': 'integer' },
'size': { 'type': 'integer' },
'sn': { 'type': 'integer' },
'text': {
    'type': 'text',
    "fielddata": True,

    # Add and search both split word by chinese
    'analyzer': 'ik_smart',
    'search_analyzer': 'ik_smart',
},
'time': { 'type': 'integer' },
'userid': {
    'type': 'text',
    "fielddata": True
},
```

4. Kibana

   在分析資料的部分，我使用了 Kibana 的軟體，不但可以用 kql 進行跟進階的搜尋，也可以 visualize 我們想看到的結果，我以文字雲做範例，首先將雜質先過濾，濾掉字元數小於二的彈幕，並且將 www, rrr, 555, qqq 等無意義彈幕篩掉，從這個圖中就可以很清楚的知道，哪些詞是最常用的。

Conclusion

從這個作業中，我學到了資料抓取，還有如何呈現資料的方式，從這些資料就能分析出很多的結果，例如：

a. user 在動畫什麼時間發最多彈幕，那個時間點可能就是精華片段

b. 找出片頭及片尾的時間點

Reference

1. https://ithelp.ithome.com.tw/articles/10252165
2. https://stackoverflow.com/questions/58462560/tag-cloud-with-text-field-in-kibana
3. http://www.voycn.com/article/docker-compose-kuaisubushu-elk-jipeizhiikfenciqi
4. https://docs.scrapy.org/en/latest/topics/request-response.html
5. https://discuss.elastic.co/t/how-to-query-for-length-with-elasticsearch-kql/259815/13
6. https://ithelp.ithome.com.tw/articles/10239765
7. https://atceiling.blogspot.com/2018/05/elk4kibana.html
8. https://levelup.gitconnected.com/docker-compose-made-easy-with-elasticsearch-and-kibana-4cb4110a80dd