

NYCU Generative Artificial Intelligence

Homework 4 - Crawling

繳交期限: 2023/05/05 23:59

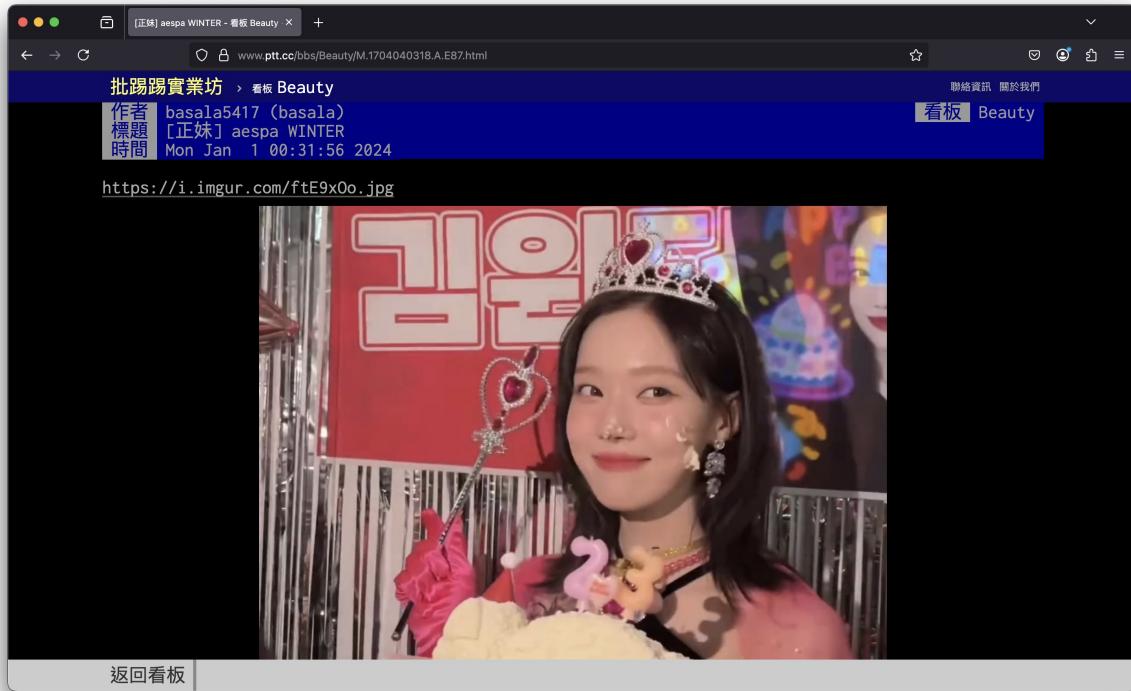
TA Hour: 週四 15:00~16:30

TA Email: yilun.ee08@nycu.edu.tw

目標

爬PTT Beauty板2024一整年的文章，2024年的第一篇文章為 [正妹] aespa WINTER

<https://www.ptt.cc/bbs/Beauty/M.1704040318.A.E87.html>。



需要繳交一份 python 腳本，檔名為 `{student_id}.py`，支援四種命令列介面功能：

- Crawl
- Push
- Popular
- Keyword

[25%] Crawl

- 格式：

```
$ python {student_id}.py crawl
```

範例：

```
$ python 0850726.py crawl
```

- 功能：

- 爬2024年所有文章。
- 忽略標題含有 [公告] 和 Fw:[公告] 的文章。
- 忽略缺少標題、標題是空字串的文章。
- 忽略沒有對應網址的文章。
- 列表頁面顯示的文章標題可能會和內文顯示的標題有差異，在這裡以列表頁面的為準。

- 輸入：

沒有輸入。

- 輸出：

- 格式說明：

在當前資料夾輸出兩個檔案：

- articles.jsonl

包含所有文章。

- popular_articles.jsonl

包含所有推爆的文章。

兩個檔案的格式均為 jsonlines，其中每一個 json 代表一篇文章的資訊，文章不需要按照日期排序，其json格式為：

```
{"date": "{文章日期}", "title": "{文章標題}", "url": "{文章網址}"}
```

- 範例 articles.jsonl 的前幾行：

```
{"date": "0101", "title": "[正妹] aespa WINTER ", "url": "https://www.ptt.cc/bbs/General/M.1683451311.A.0101.I.00000000000000000000000000000000"}, {"date": "0101", "title": "[帥哥] 言承旭", "url": "https://www.ptt.cc/bbs/General/M.1683451311.A.0101.I.00000000000000000000000000000000"}, {"date": "0101", "title": "[正妹] 2024台北車展SG (補君白照)", "url": "https://www.ptt.cc/bbs/General/M.1683451311.A.0101.I.00000000000000000000000000000000"}
```

```
{"date": "0101", "title": "[正妹] 許允真", "url": "https://www.ptt.cc/bbs/B..."}  
...
```

文章順序不會影響評分。

文章的 url 請以 https:// 開頭。

第一個測試指令一定是 crawl，後續所有測試都可以讀取 articles.json 和 popular_articles.json。

- 日期、標題和URL：

批踢踢實業坊 > 看板 Beauty

看板 | 個人區

搜尋文章...

推文數 - 嘘文數

忽略公告

標題

日期

序號	標題	日期
30	[正妹] aespa Karina wafiea708	1/01 ...
	[公告] 水桶 lude71 soulKnight	1/01 ...
X6	[帥哥] 二兵 袁育成 suchungzen52	1/01 ...
	[公告] 水桶 yokann night	1/02 ...
7	[正妹] Cosplay 048 日本 換裝娃娃 Gentlemon	1/02 ...
9	[正妹] 葵鑽子 gogo94	1/02 ...
19	Ella Netzer 以色列女孩 Kjartan	1/02 ...
	[正妹] 馬甲 hateOnas	1/02 ...
4	[正妹] 李世榮 ashillel	1/02 ...
19	[正妹] Lia ReIKurumiya	1/02 ...
4	[正妹] 雪 spernjuice	1/02 ...
39	[正妹] 女女惑 asxc530530	1/02 ...
28	[正妹] 太妍 wafiea708	1/02 ...
6	[正妹] Lisa Matthews 美國模特兒 175cm Kjartan	1/02 ...



- 推爆定義

一篇文章被推文時，文章前會顯示推文（低調推除外）的次數，但如果被噓文時，會抵銷掉推文的次數，而推文-噓文大於100時，就會變成「推爆」的狀況。但進入爆的狀況之後，系統就不會再計算推文數，而是持續以100來計數，因此即使推文數遠超100，但這時有人噓文時，數字還是會恢復到99，這時要再有人推才會再變成爆的狀態。（或可稱為「再推爆一次」）



[25%] Push

基於 **Crawl** 找到的文章進行操作，所有被 **Crawl** 忽略的文章不在其他功能的作用範圍內。

- 格式：

```
$ python {student_id}.py push {start_date} {end_date}
```

範例：

```
$ python 0850726.py push 0304 1231
```

- 功能：

找出在 `{start_date}` (含) 跟 `{end_date}` (含) 之間的以下資訊：

- 推文和虛文兩種各自的總數。
- 推文最多次的前10名 `user_id`。
- 虛文最多次的前10名 `user_id`。

- 輸入：

- `{start_date}`、`{end_date}`

格式為MMDD，例如3/4為 `0304`，12/31為 `1231`。

- 輸出：

- 在當前資料夾輸出一個json檔，檔名請按照以下格式：

`push_{start_date}_{end_date}.json`

範例：

`push_0304_1231.json`

- json格式：

```
{
    "push": {
        "total": {推文總數},
        "top10": [
            {"user_id": "{user id}", "count": {推文數}},
            {"user_id": "{user id}", "count": {推文數}},
            ...
        ]
    },
    "boo": {
        "total": {虛文總數},
        "top10": [
            {"user_id": "{user id}", "count": {虛文數}},
            {"user_id": "{user id}", "count": {虛文數}},
            ...
        ]
    }
}
```

`top10` 是依照 `count` 由大到小排序，如果 `count` 相同，則 `user_id` 字典序 (Lexicographical Order) 較大者排序在前。如果不滿10人則只需列出僅有的人。

- 輸出範例：

```

{
    "push": {
        "total": 1040,
        "top10": [
            {"user_id": "maxxxxxx", "count": 6},
            {"user_id": "Krishna", "count": 6},
            {"user_id": "yggyyygy", "count": 5},
            {"user_id": "tyrande", "count": 5},
            {"user_id": "monarch0301", "count": 5},
            {"user_id": "johnwu", "count": 5},
            {"user_id": "cityhunter04", "count": 5},
            {"user_id": "adamlovedogc", "count": 5},
            {"user_id": "abellea85209", "count": 5},
            {"user_id": "Lailungsheng", "count": 5}
        ]
    },
    "boo": {
        "total": 247,
        "top10": [
            {"user_id": "QVQ9487", "count": 6},
            {"user_id": "theclgy2001", "count": 4},
            {"user_id": "cczoz", "count": 4},
            {"user_id": "zss40401", "count": 3},
            {"user_id": "cityhunter04", "count": 3},
            {"user_id": "yushenglu", "count": 2},
            {"user_id": "un94su3", "count": 2},
            {"user_id": "srmember", "count": 2},
            {"user_id": "sion1993", "count": 2},
            {"user_id": "saw6904", "count": 2}
        ]
    }
}

```

- 推文、噓文、中立留言說明

中立留言

編輯: magic542 (218.35.2.144 臺灣), 01/03/2023 09:45:59

推 BBWAS: 我以為她要唱變神
推 bettys111: 真的有點微妙
推 s610052003: 比之前辯子頭好太多了
→ dingading: https://youtu.be/PKEmgy6_qLE

223.137.236.143 01/03 09:45
218.164.140.153 01/03 11:09
223.138.129.229 01/03 11:41
114.45.15.140 01/03 13:28

推文

推 chemistryxi: 化妝技術越來越好
推 paris27xi: 再美也是美環 (握手)
推 qingge: 再推很會唱
推 linsred1006: 想帶他吃麥當勞
噓 un94su3: 美在哪?
噓 marchmaymay: 蛤
→ OGCA18: 噗舌青?
噓 pase139: 啪裡張鳳書了?
噓 jihheng: 塑膠 充氣娃娃是不是

39.14.50.129 01/03 13:28
36.236.171.197 01/03 14:26
106.64.154.180 01/03 15:47
49.216.225.54 01/03 18:46
111.254.217.82 01/03 20:02
114.136.189.36 01/04 12:09
49.216.24.177 01/05 12:48
220.143.30.249 01/10 01:15
39.9.137.188 01/12 15:00

推文自動更新已關閉

本網站已依台灣網站內容分級規定處理。此區域為限制版，未滿十八歲者不得瀏覽。

返回看板

[25%] Popular

基於 **Crawl** 找到的文章進行操作，所以被 **Crawl** 忽略的文章不在其他功能的作用範圍內。

- 格式：

```
$ python {student_id}.py popular {start_date} {end_date}
```

範例：

```
$ python 0850726.py popular 0304 1231
```

- 功能：

找出在 `{start_date}` (含) 跟 `{end_date}` (含) 之間的以下資訊：

- 推爆的文章數量。
- 推爆文章中的所有圖片URL，包括內文和留言中的圖片URL。
- 圖片URL定義：開頭必須是 `http://` 或 `https://`，並且要以
`.jpg`、`.jpeg`、`.png`、`.gif` 為副檔名結尾，副檔名不限大小寫。



- 輸入：

- {start_date} 、 {end_date}

格式均為MMDD，例如3/4為 0304，12/31為 1231。

- 輸出：

- 在當前資料夾輸出一個json檔，檔名請按照以下格式：

```
popular_{start_date}_{end_date}.json
```

範例：

```
popular_0304_1231.json
```

◦ json格式：

```
{
    "number_of_popular_articles": {推爆數},
    "image_urls": [
        "{url_1}",
        "{url_2}",
        ...
    ]
}
```

◦ 輸出範例：

```
{
    "number_of_popular_articles": 2,
    "image_urls": [
        "https://i.imgur.com/UDJQEyi.jpg",
        "https://i.imgur.com/jUrvWQM.jpg",
        "https://i.imgur.com/lU5JTIT.jpg",
        "http://i.imgur.com/spn4dNg.jpg",
        ...
    ]
}
```

評分時不比較順序。不用刪除重複的URL。

[25%] Keyword

請基於**Crawl**找到的文章在指定區間內找對應資訊，所有被**Crawl**忽略的文章同樣不用被考慮。

• 格式：

```
$ python {student_id}.py keyword {start_date} {end_date} {keyword}
```

範例：

```
$ python 0850726.py keyword 0304 1231 正妹
```

• 功能：

找出在 `{start_date}` (含) 和 `{end_date}` (含) 之間內文包含 `{keyword}` 的文章，並統計這些文章的以下資訊：

- 文章中的所有圖片URL，包括內文和留言中的圖片URL。
- 圖片URL定義：開頭必須是 `http://` 或 `https://`，並且要以
`.jpg`、`.jpeg`、`.png`、`.gif` 為副檔名結尾，副檔名不限大小寫。
- 內文範圍說明：
 - 從「作者」(含)開始到綠色的「* 發信站」(不含)之間，只要有出現 `{keyword}` 就算這篇文章包含 `{keyword}`。
 - 如果「* 發信站」不存在，則忽略這篇文章。
 - 內文標題和文章列表顯示的標題可能不同，以內文為準。
 - 內文出現的網址也在 keyword 匹配範圍內。



- 輸入：
 - `{keyword}`
保證不包含空白字元（例如 `space`、`tab` ...）。
 - `{start_date}`、`{end_date}`
日期格式均為MMDD，例如3/4為 `0304`，12/31為 `1231`。

- 輸出：
 - 在當前資料夾輸出一個json檔，檔名請按照以下格式：
`keyword_{start_date}_{end_date}_{keyword}.json`
 範例：
`keyword_0304_1231_正妹.json`
 - json格式：

```
{  
    "image_urls": [  
        "{url_1}",  
        "{url_2}",  
        ...  
    ]  
}
```

- 輸出範例：

```
{  
    "image_urls": [  
        "https://i.imgur.com/LNFIMk9.jpg",  
        "https://i.imgur.com/KpWP9Dm.jpg",  
        "http://i.imgur.com/A2LqXBB.jpg",  
        ...  
    ]  
}
```

評分時不比較順序。不用刪除重複的URL。

繳交內容

只能繳交一個 `.py` 檔，名稱為 `{student_id}.py`，請將 `{student_id}` 替換為你的學號。

測試環境

- 作業系統：Ubuntu 24.04。
- `Python 3.12.3` (`Python 3.12.x` 都相容)。
- 使用工程四館的獨立 IP 進行測試。
- 只能使用 `Python 3.12.x` 預設套件和以下套件，如果有需求可以和助教討論：

```
beautifulsoup4==4.13.4  
click==8.1.8  
html5lib==1.1  
httpx==0.28.1  
lxml==5.3.2  
pandas==2.2.3  
pyquery==2.0.1  
requests==2.32.3  
scrapy==2.12.0  
tqdm==4.67.1
```

- [Optional] 實際測試會在docker中進行，如下供參考

```
docker run --rm -it -v ${PWD}:/crawler yilun/nycu_25ss_gai_hw4:20250422 python
```

如果繳交格式有誤但手動修正後可正常執行，最後成績 $\times 0.8$ 。如果無法執行，則視為沒有繳交。

評分

- 評分方式

對 `{student_id}.py` 進行測試，除了 **Crawl** 以外，每種功能測試 5 組參數，每組參數的測試結果正確可得 5 分。

- 配分

功能	配分	時限(一組參數)	備註
Crawl	25%	20分鐘	
Push	25%	10分鐘	單筆參數日期跨度最多92天
Popular	25%	10分鐘	單筆參數日期跨度最多92天
Keyword	25%	10分鐘	單筆參數日期跨度最多92天

程式執行超過時限會被強制 kill。

結果比對

- 定義

Intersection over Union (IOU) 或稱 Jaccard index，為集合 A 和 B 的重疊度：

$$\text{IOU}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- Crawl**

兩筆 json 資料相同定義為日期、標題、網址三者同時相同。當你找到的文章和標準答案中的文章 $\text{IOU} > 0.95$ 視為正確

- Push**

考慮

$$S_{\text{push}} = \{s_1, s_2, \dots, s_{10}\}$$

依序為推文 top10 的 `userid` 標準答案。並且

$$S'_{\text{push}} = \{s'_1, s'_2, \dots, s'_{10}\}$$

為你的答案。同理可推 S_{boo} 和 S'_{boo} 。

定義順序數對集合為：

$$P(S) = \{(s_i, s_j) \mid \forall i < j \text{ and } s_i, s_j \in S\}$$

當以下條件滿足時

$$\frac{\text{IOU}(P(S_{\text{push}}), P(S'_{\text{push}})) + \text{IOU}(P(S_{\text{boo}}), P(S'_{\text{boo}}))}{2} > 0.8$$

視為正確。

評分時不會使用到 `total` 和 `count`，僅方便除錯使用。

• Popular

假設答案的 `image_urls` 形成集合 A ，你找到的 `image_urls` 形成集合 B ，當

$$\text{IOU}(A, B) > 0.95$$

時視為正確。

評分時不會使用到 `number_of_popular_articles`，僅方便除錯使用。

• Keyword

假設答案的 `image_urls` 形成集合 A ，你找到的 `image_urls` 形成集合 B ，當

$$\text{IOU}(A, B) > 0.95$$

時視為正確。

比對工具

壓縮檔 `hw4.zip` 中提供一個檢測腳本 `eval.py` 和一個資料夾 `2023_answer` 含有 2023 年的測試答案，可以從檔案名稱看出測試參數。假設你的輸出檔案都存在資料夾 `outputs` 中，資料夾結構如範例如下：

```
./  
└── 2023_answer  
    ├── articles.jsonl  
    ├── keyword_0501_0531_正妹.json  
    ├── keyword_0701_0731_IG.json  
    ├── keyword_0815_0910_PiTT.json  
    ...  
    |  
    └── outputs  
        ├── articles.jsonl  
        ├── keyword_0501_0531_正妹.json  
        ├── keyword_0701_0731_IG.json  
        ├── keyword_0815_0910_PiTT.json  
        ...  
└── eval.py
```

可以用以下指令測試正確性：

```
python eval.py 2023_answer outputs
```

提醒

- `Crawl` 建議用 `append` 模式開檔，不斷寫入已經爬取的文章資訊，避免遇到 `Exception` 時完全沒有儲存結果。
- 多行程 (`multi-process`) 請用 `os.cpu_count()` 取得當前邏輯處理器數量，並對行程數適當調整
- 測試結果可能因為有人推文、噓文、刪除文章等行為隨著時間改變，因此
 1. `2023_answer` 在你測試的時候不一定會是正確答案
 2. 作業批改時，答案生成和測試會盡量在短時間內完成，測試後會公布測試資料和答案。