B0928022 杜云驊

Colab-link: https://colab.research.google.com/drive/1q4XG_TBgLN2U8nvAfrMk-xdhjYz3LFrU?usp=sharing

```
import requests
import json
import jieba
import networkx as nx
import re
from bs4 import BeautifulSoup

#•15064
MOVIE_TOTAL_NUM = 15064
LINK = "https://movies.yahoo.com.tw/movieinfo_main/"

```

```
class MovieInfo:
    def __init__(self, doc_id=0, cname='unknown', ename='unknown', pagerank=0, label=['unknown'], intro='unknown', re
                 links=[]) -> None:
        self.doc_id = doc_id
        self.cname = cname
        self.ename = ename
        self.pagerank = pagerank
        self.label = label
        self.intro = intro
        self.released_date = released_date
        self.links = list(set([14653, 13733, 14652, 14690, 14848, 14301, 14558, 14822, 14850, 14557, 14208,

    def generate_dict(self) -> dict:
        dict = {}
        dict['doc_id'] = self.doc_id
        dict['cname'] = self.cname
        dict['ename'] = self.ename
        dict['pagerank'] = self.pagerank
        dict['label'] = self.label
        dict['intro'] = self.intro
        dict['released_date'] = self.released_date
        dict['links'] = self.links

        return dict

```

```
class SearchEngine:
    def __init__(self, inverted_index, path) -> None:
        self.inverted_index = inverted_index
        self.movie_data = []
        self._get_movie_data(path)

    def query(self, query) -> None:
        seg = jieba.cut_for_search(query)
        words = {word for word in seg}
        match_resaults = set(self.inverted_index[words.pop()])
        while words:
            match_resaults.intersection_update(self.inverted_index[words.pop()])

        match_resaults = self._sort_by_rankpage(match_resaults)

        if match_resaults == set():
            print("None")
        self._print_query_resault(match_resaults, query)

    def _get_movie_data(self, path) -> None:
        with open(path) as f:
            self.movie_data = json.load(f)

    def _print_query_resault(self, id_list, query) -> None:
        print(f"您的搜尋結果:")
        print(f"共{len(id_list)}筆, 符合\"{query}\"")

        regex = re.compile(rf'(.*){query}')

```

```
        relate = 0
        for movie in self.movie_data:
            if regex.match(movie['cname']) or regex.match(movie['ename']) or regex.match(movie['intro']):
                relate += 1

        count = 0
        for id in id_list:
            for movie in self.movie_data:
                if movie['doc_id'] == id:
                    if regex.match(movie['cname']) or regex.match(movie['ename']) or regex.match(movie['
                        count += 1
                    print(f"\n{id} ({movie['pagerank']}):")
                    print(f"Chinese Name: {movie['cname']}")
                    print(f"English Name: {movie['ename']}")
                    print(f"Intro: {movie['intro']}")
        print(f"\nPrecision = {count} / {len(id_list)} = {count / len(id_list) * 100}%")
        print(f"Recall = {count} / {relate} = {count / relate * 100}%")

    def _sort_by_rankpage(self, matches) -> list:
        dict = {}
        for id in matches:
            for movie in self.movie_data:
                if movie['doc_id'] == id:
                    dict[id] = movie['pagerank']
                    break

        return [key for key, pagerank in sorted(dict.items(), key=lambda x: x[1], reverse=True)]
```

```
response = requests.get("https://movies.yahoo.com.tw/chart.html")
soup = BeautifulSoup(response.text, "lxml")

resault = soup.find('div', 'rank_list table rankstyle1').find('dl', 'rank_list_box').find('h2').text.strip()
lt1 = [resault]
resault = soup.find('div', 'rank_list table rankstyle1').find_all('div', 'rank_txt')[:9]
for m in resault:
    lt1.append(m.text.strip())

response = requests.get("https://movies.yahoo.com.tw/chart.html?cate=us")
soup = BeautifulSoup(response.text, "lxml")

resault = soup.find('div', 'rank_list table rankstyle1').find('dl', 'rank_list_box').find('h2').text.strip()
lt2 = [resault]
resault = soup.find('div', 'rank_list table rankstyle1').find_all('div', 'rank_txt')[:9]
for m in resault:
    lt2.append(m.text.strip())

response = requests.get("https://movies.yahoo.com.tw/chart.html?cate=trailer")
soup = BeautifulSoup(response.text, "lxml")

resault = soup.find('div', 'rank_list table rankstyle3').find('dl', 'rank_list_box').find('h2').text.strip()
lt3 = [resault]
resault = soup.find('div', 'rank_list table rankstyle3').find_all('div', 'rank_txt')[:9]
for m in resault:
    lt3.append(m.text.strip())

ranking = set(lt1 + lt2 + lt3)
ranking.remove('做工的人電影版')

print(ranking)
```

```
{'做工的人 電影版', '疫起', '龍與地下城：盜賊榮耀', '靈魂伴侶', '金牌拳手3', '驚聲尖叫6', '魔女宅急便', '捍衛任務4', '蟻人與黃蜂女：量子狂熱', '
```

```
movie_infos = []
n = 0

for i in range(MOVIE_TOTAL_NUM):
    link = LINK + str(i + 1)
    links = []
    labels = []

    response = requests.get(link)
    soup = BeautifulSoup(response.text, "lxml")
```

```
11
12          try:
13                  resault  =  soup.find('div',  'movie_intro_info_r')
14                  cname  =  resault.find('h1').text.strip()
15                  ename  =  resault.find('h3').text.strip()
16                  released_date  =  resault.find('span',  class_=None).text[5:]
17
18                  resault  =  soup.find('div',  'level_name_box').find_all('div',  'level_name')
19                  for  label  in  resault:
20                          labels.append(label.text.strip())
21
22                  resault  =  soup.find('div',  'gray_infobox_inner')
23                  intro  =  str(resault.select_one('span').text).strip().replace('\n',  '').replace('\r',  '').replace('  ',  ' ')
24
25                  try:
26                          resault  =  soup.find('ul',  'maylike_list  starlist').find_all('li',  'gabtn')
27                          regex  =  re.compile(r'[0-9]+$')
28                          for  movie  in  resault:
29                                  links.append(int(regex.findall(movie.find('a')['href'])[0]))
30                  except:
31                          pass
32          except:
33                  continue
34
35          movie_info  =  MovieInfo(doc_id=i+1,  cname=cname,  ename=ename,  label=labels,  intro=intro,  released_date=released_dat
36          movie_infos.append(movie_info)
37          n  +=  1
38
39 print(n)
```

```
11450
```

```
1 G  =  nx.DiGraph()
2 for  movie_info  in  movie_infos:
3          for  link_target  in  movie_info.links:
4                  G.add_edge(movie_info.doc_id,  link_target)
5 pagerank_list  =  nx.pagerank(G,  alpha=1)
6 sorted_pagerank_list  =  sorted(pagerank_list.items(),  reverse=True)
7 for  movie_info  in  movie_infos:
8          item  =  sorted_pagerank_list.pop()
9          movie_info.pagerank  =  item[1]
10
11 movie  =  [movie_info.generate_dict()  for  movie_info  in  movie_infos]
12
13 with  open("hw2.json",  "w")  as  f:
14          json.dump(movie,  f,  indent=4)
15
```

```
1 inverted_index  =  {}
2 for  movie  in  movie_infos:
3          seg  =  jieba.cut_for_search(movie.cname)
4          for  word  in  seg:
5                  if  word  in  inverted_index.keys():
6                          if  movie.doc_id  in  inverted_index[word]:
7                                  continue
8                          inverted_index[word].append(movie.doc_id)
9                  else:
10                         inverted_index[word]  =  [movie.doc_id]
11
12         seg  =  jieba.cut_for_search(movie.intro)
13         for  word  in  seg:
14                 if  word  in  inverted_index.keys():
15                         if  movie.doc_id  in  inverted_index[word]:
16                                 continue
17                         inverted_index[word].append(movie.doc_id)
18                 else:
19                         inverted_index[word]  =  [movie.doc_id]
20
21 with  open("inverted.json",  "w")  as  f:
22         json.dump(inverted_index,  f,  indent=4)
23
```

```
with  open("inverted.json")  as  f:
        inverted_index  =  json.load(f)

search_engine  =  SearchEngine(inverted_index,  "hw2.json")
search_engine.query("捍衛任務")

```

```
您的搜尋結果:
共41筆, 符合"捍衛任務"

13573 (1.5808705135843418e-06):
Chinese Name: 殺戮基地
English Name: Black Site
Intro: ★ 《怒火邊界》《捍衛任務》製片打造黑暗系火爆動作鉅片★ 《不可能的任務系列》動作女星蜜雪兒摩納漢挑大樑主演★ 《魔鬼終結者：創世契機》男

14653 (1.0647066239985156e-06):
Chinese Name: 捍衛任務4
English Name: John Wick: Chapter 4
Intro: ★ 台灣搶先全球上映，IMAX、Dolby Cinema版本同步上映★ 系列全球賣座近6億《捍衛任務系列》原班人馬打造最新史詩篇章★ 「菜問」對決「殺神」

12776 (8.687221522247683e-07):
Chinese Name: 救命緝約
English Name: The Contractor
Intro: ★《捍衛任務》《天劫倒數》製作團隊最爽動作鉅獻★《神力女超人》克里斯潘恩 《地獄》班佛斯特 《恐懼大街》吉莉安雅各布斯 大銀幕超激火拼★ 

14703 (8.292053684360614e-07):
Chinese Name: 驚爆點
English Name: Point Break
Intro: ★《危機倒數》奧斯卡金獎導演凱薩琳畢格羅執導 ★ 《捍衛任務系列》基努李維27歲嶄露頭角成名作★《鐵達尼號》《阿凡達：水之道》金獎名導詹姆

12899 (2.0322458521049444e-07):
Chinese Name: 非甜蜜生活
English Name: Three Floors
Intro: ★ 坎城影展競賽片 首映全場鼓掌十五分鐘★ 金棕櫚大師 南尼莫瑞提  睽違六年 深情力作★ 以色列動人小說改編 義大利金獎演員同台飆戲三個樓層、

14759 (1.345739939951112e-07):
Chinese Name: 殺手迴戰
English Name: Assassin Club
Intro: ★《金牌特務：金士曼起源》《天能》團隊傾力製作 火力全開★《不可能的任務7》《捍衛任務2》動作團隊玩命演出★《玩命快遞：肆意橫行》導演爽度

11346 (1.1019146165282226e-07):
Chinese Name: 一級任務
English Name: Voyagers
Intro: ★ 壓抑慾望被解封，無重力多人廝殺混戰★《鋼鐵英雄》《捍衛任務》億萬製片打造，超越想像嶄新烏托邦★《藥命效應》《分歧者》導演尼爾伯格 探

10821 (1.0303354377587279e-07):
Chinese Name: 追殺艾娃
English Name: Ava
Intro: ★《神鬼認證》《玩命關頭》幕後團隊打造暑期最強動作鉅片★《牠：第二章》潔西卡雀絲坦監製主演，魅力打造女力版《捍衛任務》★ 《紳士追殺令》

11556 (9.944835477592685e-08):
Chinese Name: 豬殺令
English Name: Pig
Intro: ★ 《仲夏魘》《怪獸與葛林戴華德的罪行》製片團隊打造豬豬版《捍衛任務》！★ 國際知名「爛番茄」影評網96%超高好評推薦★ 一切只為一頭豬！尼

7820 (8.808257703044979e-08):
Chinese Name: 死侍2
English Name: Deadpool 2
Intro: 《死侍2》不只包括從「死侍」本人萊恩雷諾斯再次親手擔綱本片監製與男主角的重要角色，更是首度好萊塢動作片名導《捍衛任務》大衛雷奇合作。而首

4825 (6.199851197712683e-08):
Chinese Name: 第七傳人
English Name: Seventh Son
Intro: 在傳說和巫術相互激盪的魔法時代，全世界僅存的聖殿騎士葛瑞戈里（奧斯卡得主傑夫布里吉 飾，《真實的勇氣》、《鋼鐵人》）周遊各地，尋找擁有神
```