In [ ]:

```
import requests
import re
from bs4 import BeautifulSoup

Y_MOVIE_URL = "https://movies.yahoo.com.tw/movieinfo_main/1"

# YOUR CODE HERE!
# IMPLEMENTIG YAHOO MOVIES CRAWLER

class MovieCrawler(object):

    def __init__(self):
        self.headers = {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
        self.movies = []
    def get_movies(self, page_url):
        res = requests.get(page_url, headers=self.headers)
        soup = BeautifulSoup(res.text, 'html.parser')
        movie_list = soup.find_all('div', class_='release_info')
        for movie in movie_list:
            # Get the Chinese name of the movie.
            ch_name = movie.find('div', class_='release_movie_name').a.text.strip()
            # Get the English name of the movie.
            en_name = movie.find('div', class_='release_movie_name').find('div', class_='
            # Get the URL of the movie detail page.
            movie_url = movie.find('div', class_='release_movie_name').a['href']
            # Get the URL of the movie detail page.
            release_date = movie.find('div', class_='release_movie_time').text.strip()
            release_date = release_date.split(' ')[-1]
            ## Get the introduction of the movie.
            intro = movie.find('div', class_='release_text').text.strip().replace('\n', '

            # Store the movie information in a dictionary and append it to the movies lis
            movie_dict = {
                'ch_name': ch_name,
                'en_name': en_name,
                'movie_url': movie_url,
                'release_date': release_date,
                'intro': intro
            }
            self.movies.append(movie_dict)
        # Find the link to the next page
#         next_page = soup.find('a', rel='next')
#         if next_page:
#             # Construct the link to the next page and recursively call the get_movies()
#             next_page_url = next_page['href']
#             self.get_movies(next_page_url)

#         return self.movies


# # DO NOT MODIFY THE VARIABLES
crawler = MovieCrawler()
movies = crawler.get_movies(Y_MOVIE_URL)

# # THE RESULTS : AS THE FOLLOWING SECTION
# # {'ch_name', 'en_name', 'movie_url', 'release_date', 'intro'}
print(len(movies))
```

```
print(*movies, sep="\n")
```

In [ ]:

```
import requests
from bs4 import BeautifulSoup

url = 'https://movies.yahoo.com.tw/movieinfo_main/1'

response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

cname = soup.find('h1', {'class': 'movie_intro_info_r'}).text.strip() if soup.find('h1',
ename = soup.find('h3', {'class': 'movie_intro_info_e'}).text.strip() if soup.find('h3',
labels = soup.find_all('div', {'class': 'level_name'})
class_labels = [label.text.strip() for label in labels] if labels else None
intro = soup.find('span', {'id': 'story'}).text.strip().replace('\n', '').replace('\u3000
released_date = soup.find('span', {'class': 'movie_intro_info_date'}).text.strip if soup.

data = {
    'cname': cname,
    'ename': ename,
    'class_labels': class_labels,
    'intro': intro,
    'released_date': released_date,
}

print(data)
```

In [ ]:

```
import requests
from bs4 import BeautifulSoup

url = 'https://movies.yahoo.com.tw/movieinfo_main/1'

response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')

cname = soup.find('div', {'class': 'movie_intro_info_r'}).find('h1').text
ename = soup.find('div', {'class': 'movie_intro_info_r'}).find('h3').text
labels = soup.find_all('div', {'class': 'level_name'})
class_labels = [label.text.strip() for label in labels] if labels else None
intro = soup.find('span', {'id': 'story'}).text.strip().replace('\n', '').replace('\u3000
released_date = soup.find('div', {'class': 'movie_intro_info_r'}).find('span').text.strip

data = {
    'cname': cname,
    'ename': ename,
    'class_labels': class_labels,
    'intro': intro,
    'released_date': released_date
}

print(data)
```

In [ ]:

```
import requests
from bs4 import BeautifulSoup

def fetch_movie_data(url,i):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')

    cname = soup.find('div', {'class': 'movie_intro_info_r'}).find('h1').text
    ename = soup.find('div', {'class': 'movie_intro_info_r'}).find('h3').text
    labels = soup.find_all('div', {'class': 'level_name'})
    class_labels = [label.text.strip() for label in labels] if labels else None
    intro = soup.find('span', {'id': 'story'}).text.strip().replace('\n', '').replace('\u
    released_date = soup.find('div', {'class': 'movie_intro_info_r'}).find('span').text.s

    data = {
        'doc_id' : id,
        'cname': cname,
        'ename': ename,
        'pagerank':pagerank,
        'class_labels': class_labels,
        'intro': intro,
        'released_date': released_date
        'link' :
    }

    return data

base_url = 'https://movies.yahoo.com.tw/movieinfo_main/'
start_id = 799
end_id = 801

for i in range(start_id, end_id + 1):
    url = base_url + str(i)
    try:
        movie_data = fetch_movie_data(url)
        print(f"Movie {i}:")
        print(movie_data)
    except Exception as e:
        print(f"Error while fetching data for movie {i}: {e}")
```

In [ ]:

```
import requests
from bs4 import BeautifulSoup
import json

def fetch_movie_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')

    cname = soup.find('div', {'class': 'movie_intro_info_r'}).find('h1').text.replace('/r
    ename = soup.find('div', {'class': 'movie_intro_info_r'}).find('h3').text
    labels = soup.find_all('div', {'class': 'level_name'})
    class_labels = [label.text.strip().replace('期待度').replace('滿意度') for label in la
    intro = soup.find('span', {'id': 'story'}).text.strip().replace('\n', '').replace('\u
    released_date = soup.find('div', {'class': 'movie_intro_info_r'}).find('span').text.s

    data = {
        'doc_id':0,
        'cname': cname,
        'ename': ename,
        'pagerank':0,
        'label': class_labels,
        'intro': intro,
        'released_date': released_date,
        'links':url
    }

    return data

base_url = 'https://movies.yahoo.com.tw/movieinfo_main/'
movies_data = []

for i in range(799, 802):
    url = base_url + str(i)
    try:
        movie_data = fetch_movie_data(url)
        movie_data['doc_id']=i
        print(f"Movie {i}:")
        print(movie_data)
        movies_data.append(movie_data)
    except Exception as e:
        print(f"Error while fetching data for movie {i}: {e}")
```

OK 抓取Yahoo電影資料 存JSON檔 (movies_data.json)

In [1]:

```
import requests
from bs4 import BeautifulSoup
import json

def fetch_movie_data(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')

    cname = soup.find('div', {'class': 'movie_intro_info_r'}).find('h1').text.replace('\r
    ename = soup.find('div', {'class': 'movie_intro_info_r'}).find('h3').text
    labels = soup.find_all('div', {'class': 'level_name'})
    class_labels = [label.text.strip() for label in labels if label.text.strip() not in [
    intro = soup.find('span', {'id': 'story'}).text.strip().replace('\n', '').replace('\r
    released_date = soup.find('div', {'class': 'movie_intro_info_r'}).find('span').text.s

    data = {
        'doc_id':0,
        'cname': cname,
        'ename': ename,
        'pagerank':0,
        'label': class_labels,
        'intro': intro,
        'released_date': released_date,
        'links':url
    }

    return data

base_url = 'https://movies.yahoo.com.tw/movieinfo_main/'
movies_data = []

for i in range(0, 15065):
    url = base_url + str(i)
    try:
        movie_data = fetch_movie_data(url)
        movie_data['doc_id']=i
        print(f"Movie {i}:")
        print(movie_data)
        movies_data.append(movie_data)
    except Exception as e:
        print(f"Error while fetching data for movie {i}: {e}")

# 將資料存成JSON檔案
with open('movies_data.json', 'w', encoding='utf-8') as f:
    json.dump(movies_data, f, ensure_ascii=False, indent=4)
```

```
chool', 'pagerank': 0, 'label': ['戰爭', '動作', '科幻', '懸疑/驚悚'], 'in
tro': '《放學後戰爭活動》是韓國OTT平台TVING播出的原創劇集，改編自韓國漫畫家河
一權創作的同名網絡漫畫，由《Frost醫生》、《臨時制先生》的成勇日導演與新人編劇尹
秀合作打造，《老婆這週要出牆》、《如此耀眼》的編劇李南奎擔任編審。劇情講述為了
對抗籠罩整個天空的奇怪生命體，高三學生們開始了「真正的戰爭」而不是入學考試戰
爭。因不明球體的入侵而面臨末日危機的地球，在歷史上最惡劣的事態中，十多歲的少年
們以槍代筆展開激烈的殊死搏鬥。', 'released_date': '(2023)', 'links': 'http
s://movies.yahoo.com.tw/movieinfo_main/15009'}
Movie 15010:
{'doc_id': 15010, 'cname': '朝鮮律師(2023)', 'ename': 'Joseon Attorney',
'pagerank': 0, 'label': ['愛情', '歷史/傳記'], 'intro': '《朝鮮律師》為韓國
MBC電視台2023年推出的古裝律政劇，改編自同名網絡漫畫，由《二十五，二十一》金勝浩
副導演與《金湯匙》的李韓俊導演共同執導，《七日的王妃》崔真英編劇執筆，演員包括
禹棹奐、苞娜、車學洸(N)。劇情描述朝鮮最優秀的百戰不敗律師「姜漢秀」，他在法律上
具備淵博知識，同時還擁有迷惑群眾的外貌，是很多人羨慕的腦性男。他為了報復導致自
己父母死亡的仇人，不斷幫助受冤枉的百姓們辯護完成法典，逐漸成為百姓們的代言人和
英雄。', 'released_date': '(2023)', 'links': 'https://movies.yahoo.com.t
w/movieinfo_main/15010'}
Movie 15011:
```

In [ ]:

```
# with open("AAA.txt", "w", encoding="utf-8") as f:
#     for result in results:
#         f.write(result + "\n")
```

In [ ]:

```
# #把文章抓進text裡
# #方法一：先把txt放進本機
# f = open('AAA.txt',encoding="utf-8")
# #一行算一個文章
# text=[]
# for line in f:
#     text.append(line)
# #測試是否讀到
# print(text[0])
```

In [ ]:

```
# import jieba
# import nltk
# from nltk.corpus import stopwords
# nltk.download('stopwords')
# stop_words = set(stopwords.words('chinese'))
# for i in range (10):
#     seg_list[i] = jieba.lcut(text[i])
# #測試分詞結果


# # 移除不必要的詞彙
# for i in range (10):
#     seg_list[i] = jieba.cut(text,cut_all=False)
#     seg_list[i] = [x for x in word_list if (x !='\t')and(x !='\n')and(x !=' ')and(x !='
#                    and(x != stop_words)and(x != '/')and(x != ' 、')and(x != ']')and(x !=

# print(seg_list[0])
```

In [ ]:

```
# import jieba
# from collections import Counter
# stop_words = [' ', '\t', '\n']
# # 載入檔案並進行分詞
# with open('AAA.txt', 'r', encoding='utf-8') as f:
#     data = f.read()
# # 移除不必要的詞彙
#     word_list = jieba.cut(data,cut_all=False)
#     word_list = [x for x in word_list if x != ('\t' and'....')]
#     word_list = [x for x in word_list if x != '\n']
#     word_list = [x for x in word_list if x != ' ']
#     word_list = [x for x in word_list if x != ' ?']
#     word_list = [x for x in word_list if x != ' 、']
#     word_list = [x for x in word_list if x != '?']
#     word_list = [x for x in word_list if x != '!']
#     word_list = [x for x in word_list if x != ' !']
#     word_list = [x for x in word_list if x != ' 。']
#     word_list = [x for x in word_list if x != '~']
#     word_list = [x for x in word_list if x != ',']
```

In [ ]:

```
# print(word_list)
```

In [ ]:

```
# import jieba
# import nltk
# from nltk.corpus import stopwords

# nltk.download('stopwords')
# stop_words = set(stopwords.words('chinese'))

# # 讀取檔案，將每一行當作一篇文章存入 text 陣列
# with open('AAA.txt', encoding="utf-8") as f:
#     text = [line.strip() for line in f]

# # 測試是否讀取到文章
# print(text[0])

# # 對每篇文章進行分詞
# seg_list = [jieba.lcut(article) for article in text]

# # 移除不必要的詞彙和符號
# filtered_list = []
# for word_list in seg_list:
#     filtered_words = [
#         x for x in word_list
#         if (x not in stop_words) and (x not in {'\t', '\n', ' ', '？', '、', '?', '!',
#     ]
#     filtered_list.append(filtered_words)

# # 測試分詞結果
# print(filtered_list[1])
```

OK 把一筆筆電影的中文名字，分類，介紹一起做分詞(一部電影算一個文件)

In [5]:

```
import json
import jieba
from collections import defaultdict
import re
import nltk
from nltk.corpus import stopwords


# 讀取資料
with open('movies_data.json', 'r', encoding='utf-8') as f:
    movies_data = json.load(f)


nltk.download('stopwords')
stop_words = set(stopwords.words('chinese'))

# 分詞並過濾中文停用詞
for movie in movies_data:
    # 只對 cname, label, intro 欄位進行分詞
    for field in ['cname', 'label', 'intro']:
        if field == 'label':
            words = jieba.cut(' '.join(movie[field]), cut_all=False)
        else:
            words = jieba.cut(movie[field], cut_all=False)
        # 過濾中文停用詞
        words = [w for w in words if w not in {stopwords,'、','\t', '\n', ' ', '？', '，'
        # 將分詞結果合併起來
        movie[field + '_words'] = ' '.join(words)

#         print(movie[field + '_words'])
# print(movie)
# print("")
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

中文分詞後，建立 Inverted Index

In [6]:

```
from collections import defaultdict

# 建立倒排索引
def build_inverted_index(movies_data, fields):
    inverted_index = defaultdict(set)

    for movie in movies_data:
        movie_id = movie['doc_id']

        for field in fields:
            words_field = field + '_words'
            words = movie[words_field].split()

            for word in words:
                inverted_index[word].add(movie_id)

    return inverted_index

# 呼叫函數，建立倒排索引
fields = ['cname', 'label', 'intro']
inverted_index = build_inverted_index(movies_data, fields)

# 顯示倒排索引
# for word, movie_ids in inverted_index.items():
#     print(f"{word}: {movie_ids}")
```

利用 PageRank 演算法來排序

In [7]:

```
import networkx as nx

# 創建有向圖
G = nx.DiGraph()

# 添加節點
for movie in movies_data:
    G.add_node(movie['doc_id'])

# 添加邊
for movie in movies_data:
    for incoming_movie_id in inverted_index[movie['cname_words']]:
        G.add_edge(incoming_movie_id, movie['doc_id'])

# 計算 PageRank
page_ranks = nx.pagerank(G, alpha=0.85)

# 按照 PageRank 值進行排序
ranked_movies = sorted(movies_data, key=lambda x: page_ranks[x['doc_id']], reverse=True)

# 印出結果
# for movie in ranked_movies:
#     print(movie['cname'], page_ranks[movie['doc_id']])

for movie in movies_data:
    movie['pagerank'] = page_ranks[movie['doc_id']]

# 將 movies_data 列表寫入 JSON 文件中
with open('movies_data_2.json', 'w', encoding='utf-8') as f:
    json.dump(movies_data, f, ensure_ascii=False, indent=4)
```

把pagerank存進原本json檔案

In [8]:

```
import json

# 讀取 movies_data_2.json 文件
with open('movies_data_2.json', 'r', encoding='utf-8') as f:
    movies_data_2 = json.load(f)

# 刪除不需要的鍵
for movie in movies_data_2:
    del movie['cname_words']
    del movie['label_words']
    del movie['intro_words']

# 將修改後的內容寫入新的 JSON 文件中
with open('movies_data.json', 'w', encoding='utf-8') as f:
    json.dump(movies_data_2, f, ensure_ascii=False, indent=4)
```

In [ ]:

```
# import networkx as nx

# # 创建有向图
# G = nx.DiGraph()
# edges = [('0', '1')]

# for i in range(1, 15064):
#     edges.append(('{}'.format(i), '{}'.format(i - 1)))
#     edges.append(('{}'.format(i), '{}'.format(i + 1)))

# # 添加边
# for edge in edges:
#     G.add_edge(edge[0], edge[1])

# # 计算 PageRank 值
# pagerank_list = nx.pagerank_scipy(G, alpha=0.85, tol=1e-6)

# # # 输出所有节点的 PageRank 值
# # print("PageRank value of all nodes: ", pagerank_list)
```

OK輸入搜尋關鍵字

In [9]:

```
import re

# 輸入搜尋關鍵字
keyword = '關頭'

# 將搜尋關鍵字變色的函數
def highlight_keyword(text, keyword):
    return re.sub(r'(' + keyword + ')', r'\033[1m\033[91m\1\033[0m', text, flags=re.IGNOR

# 搜尋符合關鍵字的電影並輸出相關資訊
b=0
a=0
doc_id_array=[]
for movie in ranked_movies:
    if keyword in movie['cname']or keyword in movie['intro']:
            b += 1
    if keyword in movie['cname_words'] or keyword in movie['intro_words']:

        doc_id = movie['doc_id']
        doc_id_array.append(doc_id)
        print(f"{movie['doc_id']} ({page_ranks[movie['doc_id']]}) 中文名稱: {highlight_key
        print(f"電影敘述: {highlight_keyword(movie['intro'], keyword)}")
        if keyword in movie['cname']or keyword in movie['intro']:
            a += 1
        print()
# print(doc_id_array)

# print(a)
# print(f"匹配关键字的电影数量：{len(doc_id_array)}")
# c=len(doc_id_array)
# print(b)
if len(doc_id_array)==0:
    print(f"Percision: 0.00%")
else:
    precision = a / len(doc_id_array)
    print(f"Percision: {precision:.2%}")


if b==0:
    print(f"recall: 0%")
else:
    recall = a / b
    print(f"recall: {recall:.2%}")
```

6》）。本片剪輯師為佩特羅史加利亞與狄倫提臣諾（《00:30凌晨密令》）。服裝設計師為奧斯卡得主珍妮畢文（《窗外有藍天》，1986年最佳服裝設計）。配樂作曲家為強伊克史崔（《毒利時代》）。《失控獵殺:第44個孩子》由雷利史考特、麥可薛佛（《出埃及記：天地王者》）與奧斯卡得主葛瑞格利夏皮洛（《危機倒數》，2009年最佳影片）聯合製作。

5834 (1.3232355756297015e-05) 中文名稱: 獵巫行動：大滅絕 英文名稱: The Last Witch Hunter
電影敘述: ★馮迪索超越從影極限，挑戰地表最強女巫獵人，再創影壇動作代表作！★《玩命關頭》製片群和《300壯士》原班團隊聯手打造年度動作鉅作！★《復仇者聯盟》特效團隊重金操刀，開創史詩冒險動作大作！世上最後一位永生不朽的女巫獵人寇特（馮迪索飾），背負著世代傳承對抗女巫的命運，在女巫與人類的世紀戰爭之中，他發現邪惡的勢力正蠢蠢欲動，將發動一場大規模的瘟疫攻擊，試圖毀滅全世界的人類，身負消滅邪惡使命重任的寇特，明白這將是他最關鍵的獵巫戰役，然而面對傾巢而出的邪惡女巫，他必須能找出扭轉戰局的關鍵，成功解救全世界……。由《玩命關頭》系列的億萬製片群，與《300壯士》原班團隊聯手打造的這部年度動作鉅作，《獵巫行動：大滅絕》是好萊塢備受期待的得獎優良原創劇本，首次搬上大銀幕，由獅門影業、頂峰娛樂聯手製作，找來商業好手才華洋溢的導演布雷克艾斯納擔任執導工作，更獲得票房巨星馮迪索的青睞，出任片中永生不朽的女巫獵人男主角寇特一角，除此之外，更集結了包括英國老牌男星米高肯恩、伊

In [10]:

```
if len(doc_id_array)==0:
    print(f"Percision: 0.00%")
else:
    precision = a / len(doc_id_array)
    print(f"Percision: {precision:.2%}")


if b==0:
    print(f"recall: 0%")
else:
    recall = a / b
    print(f"recall: {recall:.2%}")
```

```
Percision: 100.00%
recall: 96.64%
```

In [ ]:

In [ ]: