B0928002 林力

助教您好，我把電影的json檔放在雲端硬碟裡喔！因為檔案有點大，謝謝助教 link:
https://drive.google.com/drive/folders/1s6DrL2UXtwFMHcaZkw0W33kHgJJPTnSx?usp=sharing

```
!pip install selenium
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting selenium
  Downloading selenium-4.8.3-py3-none-any.whl (6.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.5/6.5 MB 36.6 MB/s eta 0:00:00
Requirement already satisfied: certifi>=2021.10.8 in /usr/local/lib/python3.9/dist-packages (from selenium) (2022.12.7)
Collecting trio-websocket~=0.9
  Downloading trio_websocket-0.10.2-py3-none-any.whl (17 kB)
Requirement already satisfied: urllib3[socks]~=1.26 in /usr/local/lib/python3.9/dist-packages (from selenium) (1.26.15)
Collecting trio~=0.17
  Downloading trio-0.22.0-py3-none-any.whl (384 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 384.9/384.9 KB 20.2 MB/s eta 0:00:00
Requirement already satisfied: exceptiongroup>=1.0.0rc9 in /usr/local/lib/python3.9/dist-packages (from trio~=0.17->selen
Requirement already satisfied: sortedcontainers in /usr/local/lib/python3.9/dist-packages (from trio~=0.17->selenium) (2.
Requirement already satisfied: sniffio in /usr/local/lib/python3.9/dist-packages (from trio~=0.17->selenium) (1.3.0)
Collecting async-generator>=1.9
  Downloading async_generator-1.10-py3-none-any.whl (18 kB)
Requirement already satisfied: idna in /usr/local/lib/python3.9/dist-packages (from trio~=0.17->selenium) (3.4)
Collecting outcome
  Downloading outcome-1.2.0-py2.py3-none-any.whl (9.7 kB)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.9/dist-packages (from trio~=0.17->selenium) (22.2.
Collecting wsproto>=0.14
  Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in /usr/local/lib/python3.9/dist-packages (from urllib3[socks]
Collecting h11<1,>=0.9.0
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 KB 3.2 MB/s eta 0:00:00
Installing collected packages: outcome, h11, async-generator, wsproto, trio, trio-websocket, selenium
Successfully installed async-generator-1.10 h11-0.14.0 outcome-1.2.0 selenium-4.8.3 trio-0.22.0 trio-websocket-0.10.2 wsp
```

```
!pip install zhon
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting zhon
  Downloading zhon-1.1.5.tar.gz (99 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.8/99.8 KB 4.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: zhon
  Building wheel for zhon (setup.py) ... done
  Created wheel for zhon: filename=zhon-1.1.5-py3-none-any.whl size=84318 sha256=ffb4820ef3d853992b691449040d7487887e4f25
  Stored in directory: /root/.cache/pip/wheels/a3/4d/f7/33026ca375a2fbdbc04f9522ac48e3f3119e6f55d4a8f38fb6
Successfully built zhon
Installing collected packages: zhon
Successfully installed zhon-1.1.5
```

```
# 爬蟲
import requests
import re
from bs4 import BeautifulSoup
# from selenium import webdriver
import time

def get_page():
  movies = []
  count = 0
  i = 15062
  while (len(movies) < 6000):
    try:
      resp = requests.get("https://movies.yahoo.com.tw/movieinfo_main/"+str(i))
    except:
      resp = None

    if resp and resp.status_code == 200:
      # print(resp.status_code)
      soup = BeautifulSoup(resp.text, 'html.parser')
      ch_list = soup.find('h1')
      # print (ch_list.text)
      en_list = soup.find('h3')
      # print (en_list.text)
      class_label = soup.find_all('div', 'level_name')
      # print (class_label[1].text)
      dates = soup.find('div', 'movie_intro_info_r')
      # print (dates.span.text)
      intros = soup.find('span', id='story')
      # print (intros.text)
```

```
        label = list()
        for j in range (len(class_label)-2):
          # print (class_label[j].text.strip())
          label.append(class_label[j].text.strip())

        if (ch_list and en_list and dates.span and intros and label):
          movie_info = {
              'doc_id': count,
              'ch_name':ch_list.text.strip(),
              'en_name':en_list.text.strip(),
              'class_label': label,
              'release_date':dates.span.text.strip(),
              'intro':intros.text.strip()
          }
          movies.append(movie_info)
          count += 1
          i -= 1
        else:
          i -= 1
          # continue
        print (count)
    return movies

# resp = requests.get("https://movies.yahoo.com.tw/category.html")
# if resp and resp.status_code == 200:
#   soup = BeautifulSoup(resp.text, 'html.parser')
#   plus = soup.find_all('div', 'movielist_info')
#   print (plus[5].text)
#   plus[5].click()
  # plus.click()
  # elem = soup.find_element_by_class_name('plus').click()
  # python_button = soup.find_elements_by_xpath("//div[@class=btn_plus_more gabtn jq-read-more-category")[0]
  # python_button.click()

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

```
串流輸出內容已截斷至最後 5000 行。
5065
5066
5066
5066
5067
5067
5067
5067
5067
5067
5068
5068
5068
5068
5069
5069
5070
5071
5071
5072
5073
5074
5074
5075
5075
5076
5076
5076
5077
5077
5077
5078
5078
5078
5078
5078
5079
5079
5079
5079
5079
5079
5080
5081
5081
5081
```

```
5081
5081
5081
5081
5081
5082
5082
5082
5083
5083
```

```
# 存成json
import json
jsObj = json.dumps(movies)
fileobject = open('movies.json', 'w')
fileobject.write(jsObj)
fileobject.close()
print (movies[1])
```

```
{'doc_id': 1, 'ch_name': '色局追兇', 'en_name': '360', 'class_label': ['愛情', '劇情', '犯罪', '懸疑/驚悚'], 'release_date':
```

```
!pip install zhon
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: zhon in /usr/local/lib/python3.9/dist-packages (1.1.5)
```

```
import requests
import re
from bs4 import BeautifulSoup
# from selenium import webdriver
import time
import jieba
from zhon.hanzi import punctuation

punctuations = [line.strip().replace('\n','') for line in open('/content/drive/MyDrive/Colab Notebooks/nlp/HW1/punctuation.txt
jieba.load_userdict('/content/drive/MyDrive/Colab Notebooks/nlp/HW1/userDict.txt')
punctuation = list(punctuation)


stopwords = [line.strip().replace('\n','') for line in open('/content/drive/MyDrive/Colab Notebooks/nlp/HW1/stopword.txt','r')
print (stopwords)
```

```
['壹', '壹.', '壹壹', '壹下', '壹個', '壹些', '壹何', '壹切', '壹則', '壹則通過', '壹天', '壹定', '壹方面', '壹旦', '壹時', '壹來',
```

```
# print (punctuations)
punctuations.append('★')
punctuations.append(',')
punctuations.append('(')
punctuations.append(')')
punctuations.append('》')
punctuations.append('《')
punctuations.append('：')
punctuations.append('。')
punctuations.append('…')
punctuations.append('、')
punctuations.append('—')
punctuations.append('；')
punctuations.append('？')
punctuations.append('！')

print (punctuation)
```

```
['"', '#', '$', '%', '&', "'", ' (', ') ', '*', '+', ', ', '-', '/', ': ', '; ', '<', '=', '>', '@', ' [', '\', '
```

```
# load json
import json
f = open('/content/drive/MyDrive/Colab Notebooks/nlp/HW2/movies.json')
data = json.load(f)
# print (data[1])
print (len(data))
# for movie in data:
  # print (movie)
f.close
clean_data = data
```

```
6000
```

```
# 去空格、換行
for movie in data:
  movie['ch_name'] = movie['ch_name'].strip()
```

```
  movie['ch_name'] = movie['ch_name'].replace(' ', '')
  movie['ch_name'] = movie['ch_name'].replace('\r', '')
  movie['ch_name'] = movie['ch_name'].replace('\n', '')
  movie['intro'] = movie['intro'].replace(' ', '')
  movie['intro'] = movie['intro'].replace('\r', '')
  movie['intro'] = movie['intro'].replace('\n', '')

print (data[1])
data_original = data
```

```
{'doc_id': 1, 'ch_name': '色局追兇', 'en_name': '360', 'class_label': ['愛情', '劇情', '犯罪', '懸疑/驚悚'], 'release_date':
```

```
# 中文分詞
import jieba

word_counts = []
for i in range (len(data)):
  data[i]['ch_name'] = jieba.lcut(data[i]['ch_name'])
  data[i]['intro'] = jieba.lcut(data[i]['intro'])

  count = {}
  for word in data[i]['ch_name']:
    if word not in punctuations and word != " " and word not in punctuation and word not in stopwords:
      if word in count:
        count[word]+=1
      else:
        count[word] = 1
  for word in data[i]['intro']:
    if word not in punctuations and word != " " and word not in punctuation and word not in stopwords:
      if word in count:
        count[word]+=1
      else:
        count[word] = 1
  word_counts.append(count)

# data[1]['ch_name'] = jieba.lcut(data[1]['ch_name'])
# data[1]['intro'] = jieba.lcut(data[1]['intro'])
print (data[0:10])
#
# print (data[1]['ch_name'])
# for movie in data:
```

```
[{'doc_id': 0, 'ch_name': ['絕命', '控制'], 'en_name': 'Control', 'class_label': ['科幻', '懸疑/驚悚'], 'release_date': '上映
```

```
# test
print (len(word_counts))
for i in range(1, 10):
  print (word_counts[i])
```

```
6000
{'色局': 3, '追': 3, '兇': 3, '奧斯卡': 4, '影帝': 3, '安東尼': 2, '霍普金斯': 2, '瑞秋懷茲': 2, '裘德洛': 2, '班佛': 1, '斯特': 1
{'人選': 2, '之人': 2, '造': 2, '浪者': 2, '2023': 1, '職人劇': 2, '製作': 1, '過多': 1, '部': 1, '超高': 1, '話題': 2, '惡的':
{'流淚': 2, '悲傷': 2, '新生代': 1, '票房': 1, '男神': 1, '蔡凡熙': 1, '許光': 2, '漢': 1, '愛上': 1, '一個女孩': 1, '演繹': 1, '
{'飛鴨': 2, '向前衝': 2, '明年': 1, '春節': 1, '連假': 1, '製作': 1, '小小': 1, '兵': 1, '神偷': 1, '奶爸': 1, '歡樂': 1, '聲音':
{'深宵': 2, '閃避': 3, '球': 3, '2023': 1, '外展': 1, '社工': 1, '楊琦': 1, '周家': 1, '怡飾': 1, '主理': 1, '體育館': 3, '計畫'
{'洗髮': 1, '魔法': 1, '二合一': 1, '2023': 1, '年輕': 1, '演員': 1, '答應': 1, '拍攝': 1, '廣告': 2, '完美': 1, '生活': 1, '出現
{'長': 1, '月': 1, '燼': 2, '明': 1, '2023': 1, '講述了': 1, '一代': 1, '魔神': 1, '澹台': 1, '衡陽': 1, '宗': 1, '掌門': 1, '
{'恩愛': 2, '兩不': 3, '疑': 3, '2023': 1, '改編自': 1, '同名': 1, '小說': 1, '該劇': 1, '講述': 1, '相看': 2, '兩相': 2, '厭':
{'藍甲': 2, '蟲': 2, '敘述': 1, '海梅': 5, '雷耶斯': 1, '剛從': 1, '大學畢業': 1, '回到': 1, '家鄉': 1, '未來': 1, '滿懷': 1, '抱
```

```
# test
all_words = []
for word in word_counts:
  all_words.extend(list(word.keys()))

print (word_counts[2])
for word in word_counts[2]:
  print (word)

print (len(word_counts))
```

```
{'人選': 2, '之人': 2, '造': 2, '浪者': 2, '2023': 1, '職人劇': 2, '製作': 1, '過多': 1, '部': 1, '超高': 1, '話題': 2, '惡的':
人選
之人
造
浪者
2023
職人劇
製作
過多
部
```

```
超高
話題
惡的
距離
做工
的人
大慕
影藝
公共電視
出品
金鐘
神劇
導演
林君陽
合作
卡司
陣容
包含
57
屆
新科
視后
謝盈
萱
黃健
瑋
王淨
陳
妍
霏
領銜主演
黃金
化身
公正
黨
幕僚
團隊
集思廣益
打出
漂亮
選戰
幕前幕後
打造
出台
灣
首部
職人
```

```
# 創造inverted index
inverted_index = {}
for i in range(len(word_counts)):
  for word in word_counts[i]:
    if word in inverted_index:
      inverted_index[word].append(i)
      # print(word)
    else:
      inverted_index[word]=list()
      inverted_index[word].append(i)
      # print(word)

    # word_occurrence[word].append(i)
# for word in all_words:
#   if word in word_occurrence:
#     word_occurrence[word].append
#   else:
#     word_occurrence[word] = 1

print (len(inverted_index))
print (inverted_index['葉問'])
print (len(inverted_index['葉問']))
print ( inverted_index['葉問'][1])
# n +1 len-1
```

```
94741
[235, 398, 3144, 3954, 4265, 4960, 5268, 5368, 5789, 5954]
10
398
```

```
# pagerank
import networkx as nx

G = nx.DiGraph()
for value in inverted_index:
  length = len(inverted_index[value])
  for count in range(length):
    for index in range(count, length):
```

```
        G.add_edge(inverted_index[value][count], inverted_index[value][index])
    # print(inverted_index[value][0])
# print (inverted_index)
pagerank_list = nx.pagerank(G, alpha=1)
print ("pagerank value: \n", pagerank_list)
```

```
pagerank value:
 {0: 1.5059331243220425e-79, 40: 6.653183858171246e-63, 173: 4.769895759624364e-51, 287: 3.4589938744121644e-47, 294: 9.2
```

```
print (len(pagerank_list))
```

```
6000
```

```
data1_original = clean_data
```

```
data_original = data1_original
```

```
# 加上 pagerank、link
for index in range(0, 6000):

  link = list()
  for key in word_counts[index].keys():
    # print (key)

    # print (inverted_index[key])
    for i in inverted_index[key]:
      if i not in link:
        link.append(i)
      # print (i)

  # print (sorted(link))
  data_original[index]= {
      'doc_id': data_original[index]['doc_id'],
      'ch_name': data_original[index]['ch_name'],
      'en_name': data_original[index]['en_name'],
      'pagerang': pagerank_list[index],
      'class_label': data_original[index]['class_label'],
      'intro' : data_original[index]['intro'],
      'release_date': data_original[index]['release_date'],
      'link': link
  }
  print (index)
# data_original[1]+{'link': 'ji'}

print (data_original[0])
# print (word_counts[1].keys())
```

```
串流輸出內容已截斷至最後 5000 行。
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
```

```
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
```

```
# test
movie_full = data_original
for i in range(5990,6000):
  print (movie_full[i])
```

```
{'doc_id': 5990, 'ch_name': '野蠻遊戲：瘋狂叢林', 'en_name': 'Jumanji: Welcome to the Jungle', 'pagerang': 1.413234452856978
{'doc_id': 5991, 'ch_name': '氣象戰', 'en_name': 'Geostorm', 'pagerang': 2.2039137377344135e-06, 'class_label': ['劇情'],
{'doc_id': 5992, 'ch_name': '美狄亞英國國家劇院現場', 'en_name': 'Medea (National Theatre Live)', 'pagerang': 3.414375977580
{'doc_id': 5993, 'ch_name': '好想大聲說出心底的話一真人版', 'en_name': 'The Anthem of the Heart', 'pagerang': 5.6306658700080
{'doc_id': 5994, 'ch_name': '叛逆的麥田捕手', 'en_name': 'Rebel in the Rye', 'pagerang': 1.0114824701706375e-05, 'class_lab
{'doc_id': 5995, 'ch_name': '相愛相親', 'en_name': 'Love Education', 'pagerang': 2.059602674598738e-05, 'class_label': ['劇
{'doc_id': 5996, 'ch_name': '狂獸', 'en_name': 'The Brink', 'pagerang': 5.133302288745145e-05, 'class_label': ['動作', '劇
{'doc_id': 5997, 'ch_name': '追捕', 'en_name': 'Man Hunt', 'pagerang': 0.00018749155602175563, 'class_label': ['動作', '劇
{'doc_id': 5998, 'ch_name': '低壓槽', 'en_name': 'The Trough', 'pagerang': 0.0018032776565804452, 'class_label': ['動作'],
{'doc_id': 5999, 'ch_name': '驅魔失控', 'en_name': 'The Possession Experiment', 'pagerang': 0.9979097481834384, 'class_lab
```

```
# 存成json
jsObj = json.dumps(movie_full)
fileobject = open('movies_full.json', 'w')
fileobject.write(jsObj)
fileobject.close()
```

```
# load json
import json
f = open('/content/drive/MyDrive/Colab Notebooks/nlp/HW2/movies_full.json')
full_movie = json.load(f)
# print (data[1])
print (len(full_movie))
# for movie in data:
  # print (movie)
f.close
```

```
6000
<function TextIOWrapper.close()>
```

```
# 搜尋引擎
class search_engine(object):

  def __init__(self):
   index = inverted_index

  def query(self, x):
    result = inverted_index[x]
    result_pagerank = {}
    j =  0
    doc_id = []
    pagerank = []
    name = []
    intro = []
    for i in result:
      doc_id.append(full_movie[i]['doc_id'])
      pagerank.append(full_movie[i]['pagerang'])
      name.append(full_movie[i]['ch_name'])
      intro.append(full_movie[i]['intro'])

    for i in range(len(doc_id)-1):
      for j in range(j, len(doc_id)-i-1):
        if pagerank[j] > pagerank[j+1]:
          temp_pagerank = pagerank[j]
          pagerank[j] = pagerank[j+1]
```

```
            pagerank[j+1] = temp_pagerank

            temp_id = doc_id[j]
            doc_id[j] = doc_id[j+1]
            doc_id[j+1] = temp_id

            temp_name = name[j]
            name[j] = name[j+1]
            name[j+1] = temp_name

            temp_intro = intro[j]
            intro[j] = intro[j+1]
            intro[j+1] = temp_intro
        # result_pagerank[j] = {
        #     'doc_id': full_movie[i]['doc_id'],
        #     'pagerank': full_movie[i]['pagerang'],
        #     'name': full_movie[i]['ch_name'],
        #     'intro': full_movie[i]['intro']
        # }
        # j += 1
    # result_pagerank = sorted(result_pagerank)
    print (pagerank)

    print ('您的搜尋結果 (Sorting by PageRank Value):')
    print ('共', len(result),'筆, 符合"', x, '" - - - 共 indexing 6000筆電影資料')
    for i in range(len(pagerank)-1, -1, -1):
        print(doc_id[i], '(', pagerank[i], '):', name[i], intro[i])
    # print (result_pagerank)

search = search_engine()
search.query('葉問')
```

```
[2.5098157211160355e-48, 4.3645618294282117e-44, 1.9076025275907578e-25, 3.0120752285139263e-22, 5.463419100835228e-21, 4
您的搜尋結果 (Sorting by PageRank Value):
共 10 筆, 符合" 葉問 " - - - 共 indexing 6000筆電影資料
5954 ( 4.339493248647045e-09 ): 環太平洋2：起義時刻 具有大規模毀滅能力，來自異次元的巨大怪獸，以及人類為了消滅牠們而打造，由人類駕駛的超巨
5789 ( 2.1057611393729336e-12 ): 港片大排檔3 ★集合火爆動作《Mrs.K》、挑戰道德尺度《以青春的名義》、懸疑刺激《搶紅》、溫暖勵志《決戰食神》
5368 ( 7.539165252967795e-16 ): 花木蘭 ★迪士尼經典動畫《花木蘭》真人版登上大銀幕★仙女姐姐劉亦菲化身花木蘭★穿上戰袍挑戰高難度武打動作★集
5268 ( 1.8882439596250882e-16 ): 葉問外傳：張天志 ★《葉問》系列甄子丹及黃百鳴監制，電影武術大師袁和平執導，武術小生張晉突破極限，挑戰完美
4960 ( 4.587947513502785e-18 ): 葉問4：完結篇. ★聖誕跨年最強IP，葉問十週年精彩完結篇★甄子丹宗師回歸，十年傳奇最後一戰★《葉問》原班人馬
4265 ( 5.463419100835228e-21 ): 葉問4：完結篇 ★聖誕跨年最強IP，葉問十週年精彩完結篇★甄子丹宗師回歸，十年傳奇最後一戰★《葉問》原班人馬再
3954 ( 3.0120752285139263e-22 ): 宗師葉問 ★《葉問1》《葉問2》正宗演員回歸前傳！★在葉問成為一代宗師之前，不為人知的故事…★杜宇航近身肉搏
3144 ( 1.9076025275907578e-25 ): 殲獄行動 ★《紅翼行動》《2槍斃命》製片打造全新動作鉅獻★《葉問4》英國武打巨星史考特艾金斯、《狙擊生死線》
398 ( 4.3645618294282117e-44 ): 捍衛任務4 ★台灣搶先全球上映，IMAX、DolbyCinema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
235 ( 2.5098157211160355e-48 ): 血仇生死鬥 ★《終極警探系列》布魯斯威利又一全新動作鉅獻★《狙擊封鎖線》《終極警探4.0》製片打造硬漢系動作片
```

```
# test
search.query('殺神')
```

```
[2.8144927477119325e-52, 4.3645618294282117e-44, 2.188797440647195e-29, 3.394800530634975e-29, 1.8012298531074327e-23, 3.
您的搜尋結果 (Sorting by PageRank Value):
共 9 筆, 符合" 殺神 " - - - 共 indexing 6000筆電影資料
5285 ( 2.526414547401287e-16 ): 潛艦獵殺令 ★《#玩命關頭》《#全面攻佔》金牌團隊聯手打造★鐵漢男星《氣象戰》#傑瑞德巴特勒X金獎影帝#蓋瑞歐德
5055 ( 1.4034779551333396e-17 ): 捍衛任務3：全面開戰 好萊塢最性感大叔基努李維飾演「地表最強殺神系列」電影續集《捍衛任務3：全面開戰》睽違兩
4736 ( 5.531571775483573e-19 ): 夜鶯的哭聲 ★《鬼敲門》導演珍妮佛肯特驚悚大作★榮獲威尼斯影展評審團特別獎、最佳新演員獎★震撼復仇大計打造女
3706 ( 3.628494837204357e-23 ): 弒樂園 ★《阿公當家》製片團隊超狂ㄎㄧㄤ作★一本正經演幹片！尼可拉斯凱吉化身殺神爆打機械人偶★沒有最狂只有更
3621 ( 1.8012298531074327e-23 ): 惡夜殺神 ★《惡魔島》製片團隊最新火爆動作鉅獻★《絕地戰警》系列編劇打造全新女英雄★《捍衛任務》《惡靈古堡
2093 ( 3.394800530634975e-29 ): 非甜蜜生活 ★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
2100 ( 2.188797440647195e-29 ): 記憶殺神 ★《007首部曲：皇家夜總會》名導馬丁坎貝爾懸疑動作鉅獻★地表最強硬漢連恩尼遜再開殺戒提槍兇猛上陣★
398 ( 4.3645618294282117e-44 ): 捍衛任務4 ★台灣搶先全球上映，IMAX、DolbyCinema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
151 ( 2.8144927477119325e-52 ): 捍衛任務 ★賣座近6億美金動作經典《捍衛任務系列》最初的原點★好萊塢男神基努李維首度化身「殺神」約翰維克★《
```