

```

In [26]: #CRAWLING DATA FROM YAHOO MOVIES
import requests
import re
from bs4 import BeautifulSoup

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

# YOUR CODE HERE!
# IMPLEMENTIG YAHOO MOVIES CRAWLER
class MovieCrawler(object):

    def __init__(self):
        self.session = requests.Session()

    def get_movies(self, page_url):
        movies = []
        # with open(page_url, 'r') as f:
        #     contents = f.read()
        for i in range(2,3):
            response = self.session.get(page_url+str(i)+"?page="+str(i))
            response.encoding = 'utf-8'
            soup = BeautifulSoup(response.text, 'html.parser')
            movie_list = soup.find_all('div', id='content_1')
            # print(movie_list)
            for movie in movie_list:
                label_class = []
                # print(movie)
                try:
                    ch_name = movie.find('div', class_='movie_intro_info_r')
                    try:
                        ch_name.span.decompose()
                    except:
                        pass
                    ch_name = ch_name.text.strip()
                    en_name = movie.find('div', class_='movie_intro_info_r')
                    release_date = movie.find('div', class_='movie_intro_info_r')
                    labels = movie.find_all('div', class_='level_name')
                    for label in labels:
                        try:
                            lb = label.find('a', class_='gabtn').text.strip()
                        except:
                            break
                        label_class.append(lb)
                    intro = movie.find('div', class_='gray_infobox_inner').s
                    doc_id = i
                    movie_info = {
                        'doc_id': doc_id,
                        'ch_name': ch_name,
                        'en_name': en_name,
                        'label[class]': label_class,
                        'release_date': release_date,
                        'intro': intro}
                    movies.append(movie_info)
                except:
                    print(i)
                    continue
        return movies

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

```

```
# # THE RESULTS : AS THE FOLLOWING SECTION
# # {'ch_name', 'en_name', 'movie_url', 'release_date', 'intro'}
print(len(movie))
print(*movie, sep="\n")
#SAVING DATA INTO JSON FILE
```

```
1
{'doc_id': 2, 'ch_name': '玩命關頭', 'en_name': 'The Fast and the Furious',
 'label[class]': ['動作', '劇情', '犯罪', '懸疑/驚悚'], 'release_date': '2001-10-13',
 'intro': '唐米尼杜洛托是洛城街頭賽車界的老大哥，他身邊有一群忠心耿耿的手下，他白天忙著組裝高性能跑車，晚上則是開著他的愛車，動輒以一次一萬美元的賭注和別人軋車。布萊恩也渴望接受極速的挑戰，他對自己的駕駛技術很有信心，但是在旁觀者的眼中他只是一個菜鳥，他開了一輛超炫的跑車想和唐老大一較高下，也希望得到他的青睞，當比賽結束，布萊恩輸得一塌塗地之後，警方接獲風聲前來取締，布萊恩在無意間從一名心狠手辣的幫派份子強尼手中救了唐老大一命，於是他就被納入唐老大的權力核心，唐老大的妹妹蜜雅也對布萊恩產生好感，但是他們都不知道布萊恩其實是一名臥底警探。布萊恩滲入賽車圈的目的是調查一連串的卡車搶案，嫌犯都是開著跑車的蒙面人，警方和聯邦調查局希望能儘早逮到搶匪，以免卡車司機採取激烈的手段對這些搶匪進行報復行動，其中最可疑的就是唐老大和強尼。正當唐老大和強尼形成水火不相容的情勢。布萊恩和唐老大兄妹的關係卻越來越深，他不但和唐老大結為好友，更忍不住對蜜雅產生好感，但是他也同時承受來自警方和FBI的壓力，必須儘快查出誰才是真正的搶匪，他在天人交戰之際，在法律和友情之間，必須做出困難的決定。'}

```

```
In [128... #SAVING DATA INTO JSON FILE
import json
with open("./moviedata.json", "w",encoding = 'utf-8') as final:
    json.dump(listObj, final,ensure_ascii=False)
```

```
In [131... #APPENDING NEW DATA INTO FILE
import json
with open("./moviedata.json", "r",encoding='utf-8') as fp:
    listObj = json.load(fp)
    # for m in movie:
    #     listObj.append(m)
print(len(listObj))
print(listObj[1])
# with open("./hw2_1.json", "w",encoding = 'utf-8') as final:
#     json.dump(intro_all, final,ensure_ascii=False)
```

```
12238
{'doc_id': 2, 'ch_name': '玩命關頭', 'en_name': 'The Fast and the Furious',
 'label[class]': ['動作', '劇情', '犯罪', '懸疑/驚悚'], 'release_date': '2001-10-13',
 'intro': '唐米尼杜洛托是洛城街頭賽車界的老大哥，他身邊有一群忠心耿耿的手下，他白天忙著組裝高性能跑車，晚上則是開著他的愛車，動輒以一次一萬美元的賭注和別人軋車。布萊恩也渴望接受極速的挑戰，他對自己的駕駛技術很有信心，但是在旁觀者的眼中他只是一個菜鳥，他開了一輛超炫的跑車想和唐老大一較高下，也希望得到他的青睞，當比賽結束，布萊恩輸得一塌塗地之後，警方接獲風聲前來取締，布萊恩在無意間從一名心狠手辣的幫派份子強尼手中救了唐老大一命，於是他就被納入唐老大的權力核心，唐老大的妹妹蜜雅也對布萊恩產生好感，但是他們都不知道布萊恩其實是一名臥底警探。布萊恩滲入賽車圈的目的是調查一連串的卡車搶案，嫌犯都是開著跑車的蒙面人，警方和聯邦調查局希望能儘早逮到搶匪，以免卡車司機採取激烈的手段對這些搶匪進行報復行動，其中最可疑的就是唐老大和強尼。正當唐老大和強尼形成水火不相容的情勢。布萊恩和唐老大兄妹的關係卻越來越深，他不但和唐老大結為好友，更忍不住對蜜雅產生好感，但是他也同時承受來自警方和FBI的壓力，必須儘快查出誰才是真正的搶匪，他在天人交戰之際，在法律和友情之間，必須做出困難的決定。'}

```

```
In [111... import json
with open("./moviedata.json", "r",encoding = 'utf-8') as final:
    listObj = json.load(final)
```

```
In [130... for l in listObj:
    if 2 in l.values():
        print("true")
        print(l)
```

```
true
```

{'doc\_id': 2, 'ch\_name': '玩命關頭', 'en\_name': 'The Fast and the Furious', 'label[class]': ['動作', '劇情', '犯罪', '懸疑/驚悚'], 'release\_date': '2001-10-13', 'intro': '唐米尼杜洛托是洛城街頭賽車界的老大哥，他身邊有一群忠心耿耿的手下，他白天忙著組裝高性能跑車，晚上則是開著他的愛車，動輒以一次一萬美元的賭注和別人軋車。布萊恩也渴望接受極速的挑戰，他對自己的駕駛技術很有信心，但是在旁觀者的眼中他只是一個菜鳥，他開了一輛超炫的跑車想和唐老大一較高下，也希望得到他的青睞，當比賽結束，布萊恩輸得一塌塗地之後，警方接獲風聲前來取締，布萊恩在無意間從一名心狠手辣的幫派份子強尼手中救了唐老大一命，於是 he 就被納入唐老大的權力核心，唐老大的妹妹蜜雅也對布萊恩產生好感，但是他們都不知道布萊恩其實是一名臥底警探。布萊恩滲入賽車圈的目的是調查一連串的卡車搶案，嫌犯都是開著跑車的蒙面人，警方和聯邦調查局希望能儘早逮到搶匪，以免卡車司機採取激烈的手段對這些搶匪進行報復行動，其中最可疑的就是唐老大和強尼。正當唐老大和強尼形成水火不相容的情勢。布萊恩和唐老大兄妹的關係卻越來越深，他不但和唐老大結為好友，更忍不住對蜜雅產生好感，但是他也同時承受來自警方和 F B I 的壓力，必須儘快查出誰才是真正的搶匪，他在天人交戰之際，在法律和友情之間，必須做出困難的決定。'}

```
In [133... #INVERTED INDEX
import jieba
import re
from zhon.hanzi import punctuation
stopwords = [line.strip().replace('\n', '') for line in open('../hw1/stopword')]
punctuations = [line.strip().replace('\n', '') for line in open('../hw1/punct')]
jieba.load_userdict('../hw1/userDict.txt')
punctuation = list(punctuation)
```

```
In [134... intro_all = []
for content in listObj:
    intro = content['intro'].replace('\n', '').replace('\t', '').replace('\u30
    intro = re.sub('[^\u4e00-\u9fa5]', '', intro)
    intro = jieba.lcut(intro)
    Intro = []
    for word in intro:
        if word not in stopwords:
            Intro.append(word)
    Intro.insert(0, content['ch_name'])
    Intro.insert(0, content['doc_id'])
    intro_all.append(Intro)
```

```
In [136... words = {}
for intro in intro_all:
    for word in intro:
        if word in words:
            words[word].append(intro[0])
        else:
            words[word] = [intro[0]]
```

```
In [ ]: from collections import Counter
inverted_index = {}
for i, j in words.items():
    print(i)
    wordcount = Counter(j)
    list(wordcount.items()).sort(key = lambda x : x[1], reverse = True)
    wordcounts = list(wordcount.items())
    inverted_index[i] = wordcounts
```

```
In [141... #PAGERANK
import networkx as nx
G = nx.DiGraph()
Edges = []
for word, doc_id in inverted_index.items():
    edges = []
    for ID in doc_id[2:]:
```

```

if ID[0] != doc_id[0][0]:
    edges.append((doc_id[0][0], ID[0]))
Edges.append(edges)

```

```

In [143... for edges in Edges:
            for edge in edges:
                G.add_edge(edge[0], edge[1])

```

```

In [ ]: import networkx as nx
pagerank_list = nx.pagerank(G, alpha = 1)
print(pagerank_list)

```

```

In [ ]: PageRankList = list(pagerank_list.items())
PageRankList.sort(key = lambda x:x[0])

```

```

In [233... #CREATING DOC LINKS
DOC_LINK = {}
doc_link = []
i = 0
for data in inverted_index.values():
    id_num = intro_all[i][0]
    doc_id = data[0][0]
    for ID in data:
        if ID[0] not in doc_link and ID[0] != doc_id:
            doc_link.append(ID[0])
    if doc_id != id_num:
        i+=1
        doc_link.sort()
        DOC_LINK[id_num] = doc_link
        doc_link = []

```

```

In [298... #SAVING DATA into json file hw2.json
# print(dict(listObj[0]))
DATA = []
for listobj in listObj:
    try:
        d_link = DOC_LINK[dict(listobj)['doc_id']]
    except:
        d_link = "None"
    try:
        prl = dict(PageRankList)[dict(listobj)['doc_id']]
    except:
        prl = 0
    info = {
        'doc_id' : dict(listobj)['doc_id'],
        'ch_name' : dict(listobj)['ch_name'],
        'en_name' : dict(listobj)['en_name'],
        'pagerank' : prl,
        'label[class]' : dict(listobj)['label[class]]',
        'intro' : dict(listobj)['intro'],
        'release_data' : dict(listobj)['release_date'],
        'links(doc_id)' : d_link,
    }
    # print(dict(listobj)['doc_id'])
    DATA.append(info)

```

```

In [302... #turn data into dictionary with id as key
movie_data = {}
for movie in DATA:
    movie_data[dict(movie)['doc_id']] = movie

```

# QUERY

```
In [440... class search_engine():
    def query(text):
        count_all = 0
        count = 0
        movie_prl = []
        movie_list = []
        #find all movie related to text
        regex = re.compile(rf'(.*){text}')
        for movie,data in movie_data.items():
            if regex.match(dict(data)['ch_name']) or regex.match(dict(data)[
                count_all += 1
        for movie,index in inverted_index.items():
            if str(text) in str(movie):
                for i in index:
                    movie_list.append(i[0])
        #print all movies related with text
        movies_list=list(dict.fromkeys(movie_list))
        data = []
        for ID in movies_list:
            count+=1
            data.append([movie_data[ID]['doc_id'],movie_data[ID]['pagerank'])
        data.sort(key = lambda x:x[1],reverse = False)
        for movie in data:
            print("DOC_ID:",movie[0])
            print("CHINESE TITLE:",movie[2])
            print("ENGLISH TITLE:",movie[3])
            print("PAGERANK:",movie[1])
            print("INTRO:",movie[4])
            print("\n")
        print("Precision = "+ str(count/len(movie_list)*100)+"%")
        print("Recall = "+str(count/count_all*100)+"%")
        print(count)
        print(count_all)
        print(len(movie_list))
```

```
In [441... search_engine.query("捍衛戰士")
```

DOC\_ID: 5248

CHINESE TITLE: 殲滅天際線

ENGLISH TITLE: Red Sky

PAGERANK: 6.119591928664552e-05

INTRO: ★《萬惡城市》製作群再現磅礴空戰，誓死炸翻大銀幕！麥斯是名飛行員，接獲上級任務進行殲滅行動，卻不料執行失誤，造成飛行戰隊極大損傷，麥斯也遭強迫除役。許久後，恐怖組織伺機而動，麥斯又被重招回部隊執行機密任務，卻發現他的朋友竟從昔日戰友變成敵軍。麥斯駕駛著戰鬥機，肩負著國家存亡，更為了平反個人榮譽，在浩瀚天際展開一場榮耀血戰。繼《捍衛戰士》、《空戰英豪》後，大銀幕上已經許久沒有出現空戰電影。此次由《萬惡城市》製片群再次重拾題材，斥資三億打造史詩級空戰。片中更結合當今反恐題材，讓故事更為貼近大眾。男主角肯吉甘特走紅於《暮光之城》，以《舞孃俱樂部》再次讓觀眾注意到這位帥氣小生。出眾的外型，被認為是布萊德彼特和柯林法瑞爾的合體，受到電影界的高度矚目。但31歲的他，其實十分專情，2009年與女友生下一女。攜著女兒逛大街的畫面，時常躍上雜誌報導。

DOC\_ID: 14847

CHINESE TITLE: 噩夢輓歌 4K數位修復版

ENGLISH TITLE: Requiem For A Dream

PAGERANK: 7.295333251493753e-05

寡婦莎拉（艾倫鮑絲汀 飾）夢想穿著年輕時的紅洋裝上電視，於是決定靠藥物來瘦身。與此同時，她的兒子哈利（傑瑞德雷托 飾）和朋友泰隆（馬龍韋恩斯 飾）希望透過販售海洛因致富，哈利的女友瑪麗安（珍妮佛康納莉 飾）也加入他們的計畫。這時四人還不知道，藥物將使他們的人生走向不可挽回的境地.....。

DOC\_ID: 13382

CHINESE TITLE: 蜘蛛頭監獄

ENGLISH TITLE: Spiderhead

PAGERANK: 8.060887200117889e-05

有時候，藥物能讓他們感覺良好。想讓心情好起來嗎？不如試試這款藥。覺得無言以對？我們也有好藥可用。然而，同為受試者的傑夫（Jeff）（麥爾斯·泰勒（Miles Teller）飾）與莉琪（Lizzy）（朱妮·絲莫利特（Jurnee Smollett）飾）結為好友，兩人的救贖之路也隨著阿奈斯提的實驗將自由意志逼到極限，而出現更詭譎的轉折。結合多種戲劇類型，同時展現黑色幽默的心理驚悚片《蜘蛛頭監獄》由約瑟夫·科辛斯基（Joseph Kosinski）（《創：光速戰記》（Tron: Legacy）、《捍衛戰士：獨行俠》（Top Gun: Maverick））執導，雷特·瑞斯（Rhett Reese）與保羅·韋尼克（Paul Wernick）（《惡棍英雄：死侍》（Deadpool）、《屍樂園》（Zombieland））編劇。

DOC\_ID: 14258

CHINESE TITLE: 獵空計畫

ENGLISH TITLE: Mission Sky

PAGERANK: 8.147062219154974e-05

電影取材自發生於 2015 年敘利亞內戰期間的真實事件，是第一部以敘利亞內戰期間俄國反恐行動為題材的電影，還找來事件當事人康斯坦丁穆拉維歐夫少校擔任電影的首席顧問，完美還原事件發生的經過。而為了力求真實，電影除了在真正的軍事基地取景拍攝，主要演員也特地到軍營中受訓數個月，再加上片中巧妙融合了敘利亞戰爭期間的真實錄像，以及《T-34：玩命坦克》視效團隊與《守護者聯盟》聲音指導聯手打造的頂級視聽饗宴，宛如把觀眾帶回戰場前線，體驗空軍軍官面對種種嚴峻考驗時的掙扎與毅力。

DOC\_ID: 14113

CHINESE TITLE: 極速甩尾

ENGLISH TITLE: Alivehoon

PAGERANK: 8.304584629459724e-05

內向的大羽絨一不擅長與人溝通，但他卻是個遊戲天才，更是賽車遊戲的日本第一好手。此時一支面臨解散危機的甩尾車隊看上他的能力邀請他加入，從虛擬轉移到真實的他，面對失敗就會喪命無法重來的後果，以及為此賭上性命的其他車手。在這過程中他逐漸成長，並以熱情、勇氣及團隊力量，跨越一道又一道的難關。

DOC\_ID: 13632

CHINESE TITLE: 決戰38度線

ENGLISH TITLE: Devotion

PAGERANK: 9.72965387573206e-05

真實故事同名暢銷小說改編，描述韓戰期間，美國非裔戰鬥機飛行員傑西（喬納森梅傑斯 飾）奉命加入突擊戰鬥機中隊，認識了同為戰鬥機駕駛的托馬斯（格蘭鮑威爾 飾）。在美國種族隔離制度之下，兩人發展出血濃於水的兄弟情感、一起執行超過20次戰鬥任務。一次行動中，傑西駕駛的僚機被擊落，一場驚險的救援行動即將展開……。

DOC\_ID: 13705

CHINESE TITLE: 法拉利

ENGLISH TITLE: Ferrari

PAGERANK: 0.00011002822036603912

《捍衛戰士》團隊和布萊德彼特打造F1電影後，也不遑多讓再加入影集《法拉利》編劇史蒂芬奈特非常興奮能有機會講述全球數一數二經典品牌的雋永事蹟，他表示「這部影集將會是向恩佐法拉利戲劇性的個性，走成抱負、榮耀般傲氣的一生致敬的作品。」而身為義大利人的導演斯特法諾索利瑪（Stefano Sollima）對於能講述義大利的精神代表之一感到非常榮幸，目前與團隊正在 F1 方程式經典賽道之一的義大利羅馬拍攝。

DOC\_ID: 10159

CHINESE TITLE: 捍衛戰士：獨行俠

ENGLISH TITLE: Top Gun: Maverick

PAGERANK: 0.00013106474210131686

《捍衛戰士：獨行俠》在美國贏得評論家和觀眾的一致好評，並在爛番茄上獲得97%的評論家評分和99%的觀眾評分。破湯姆克魯斯從影以來首週開片和累積票房最高紀錄，破“陣亡將士紀念日週末”影史票房紀錄；破美國派拉蒙影業影史票房紀錄；破影史第一部電影在陣亡將士紀念日週末和勞動節週末票房排名第一的電影；同時，破Skydance電影製片的電影票房最高紀錄，其次是《不可能的任務：全面瓦解》。

DOC\_ID: 7272

CHINESE TITLE: 氣象戰

ENGLISH TITLE: Geostorm

PAGERANK: 0.000137081687952072

INTRO: 在一連串前所未有的大自然災難威脅地球之後，全世界的領導人聚首，一同創造出一套複雜的衛星網路，來控制全球氣候及保護全人類的安全。但如今出了狀況，這套原本用來保護地球的系統卻反過來攻擊地球，人類要與時間賽跑，在這場全球地磁風暴毀滅一切及天地萬物之前發現真正的威脅。最新懸疑冒險片《氣象戰》是《ID4星際重生》編劇兼製片人狄恩戴夫林執導的首部電影長片，由傑瑞德巴特勒（《全面攻佔：倒數救援》、《300壯士》）、吉姆史特格斯（《雲圖：穿越過去未來》）、艾比柯妮許（《藥命效應》）、亞歷珊卓瑪麗亞羅納（《決戰終點線》）、吳彥祖（《鐵拳男子》、《魔獸：崛起》）、歐亨尼奧德伯茲（《如何成為拉丁情人》），加上奧斯卡獎提名演員艾德哈里斯（《時時刻刻》、《阿波羅13》）及安迪賈西亞（《教父3》）主演。傑瑞德巴特勒飾演傑克，是一名科學家，他接獲任務要與他弟弟麥斯一同解決這個衛星計劃的機能失常，麥斯由吉姆史特格斯飾演；艾比柯妮許飾演特務探員莎拉威爾森；亞歷珊卓瑪麗亞羅納飾演國際太空站（ISS）的太空人尤蒂法斯賓達，她負責管理太空站；吳彥祖飾演鄭先生，荷蘭男孩計劃（Dutch Boy Program）的香港監督；歐亨尼奧德伯茲飾演太空站組員赫南德茲；安迪賈西亞飾演美國總統安德魯帕瑪；以及艾德哈里斯飾演國務卿李奧納德戴肯。其他卡司包括薩琪畢茲（新片《死侍2》）、影集《亞特蘭大》、阿德普波奧杜耶（《大賣空》、《自由之心》）、阿瑪威克（《露西》、《諜對諜》）及羅伯席安（《天使聖物：骸骨之城》、《魔女神兵》）。這部電影由狄恩戴夫林與保羅吉歐特編劇，製片人是狄恩戴夫林及Skydance公司的大衛艾利森與唐娜戈柏，監製是休伯特W甘恩斯、Electric Entertainment公司的馬克羅斯金及Skydance公司的唐格蘭傑，聯合製片人是克里夫蘭寧及Electric Entertainment公司的瑞秋歐夏恩。幕後創意團隊包括攝影指導羅伯特謝佛（《尋找新樂園》、《007量子危機》）；製作設計寇克M佩楚謝利（《白宮末日》）；剪接朗羅森（《哥吉拉》）、奧斯卡獎提名剪接克里斯利賓遜（《赤色風暴》、《捍衛戰士》）和奧斯卡獎提名剪接約翰雷佛（《阿凡達》）；服裝設計蘇珊麥特森（《大賣空》、《狡兔計畫》）；以及視覺效果監督傑佛瑞A歐昆（《超世紀封神榜》、《當地球停止轉動》）。配樂則是羅恩巴夫（《樂高蝙蝠俠電影》）。《氣象戰》由華納兄弟影業與Skydance公司出品，Skydance與Electric Entertainment公司製作，華納兄弟發行。預計於2017年10月20日起在全球上映，有2D、3D及IMAX版本。

DOC\_ID: 13876

CHINESE TITLE: 晨間直播秀

ENGLISH TITLE: The Morning Show

PAGERANK: 0.00015440061737114212

多年飾演廣告創意總監的喬漢姆，將把他七年扮演事業有成人物、卻又充滿不安與步步為營的惟妙惟肖樣貌，帶入《晨間直播秀》第三季，在劇中飾演事業有成的財團巨頭保羅馬克，試圖運用手段拉攏兩位

主播和電視台總監比利古達達到收購 UBA 電視台的目的。

DOC\_ID: 14585  
CHINESE TITLE: 迫降危機  
ENGLISH TITLE: Plane  
PAGERANK: 0.0002587925106771915  
客機機長布羅迪（傑瑞德巴特勒 飾）在一次航行中因為遇到風暴，而不得不緊急迫降，結果卻意外降落在被武裝份子佔據的菲律賓霍洛島，機上乘客隨即遭到歹徒挾持，出於身為機長的責任感，布羅迪決定和機上移送的謀殺犯路易斯（麥克寇特 飾）合作，無論如何都要救出乘客並把他們平安送回家.....。

DOC\_ID: 13310  
CHINESE TITLE: 逃出封鎖線  
ENGLISH TITLE: The Pilot  
PAGERANK: 0.0002658320317511211  
《逃出封鎖線》是由真人實事改編，故事靈感來自於二戰時期的一位飛行員，英勇奮戰努力從敵軍領土成功逃亡回家的傳奇故事。電影中冒險刺激的空戰場面媲美《捍衛戰士》，強大堅定的生存意志比擬《127小時》，絕對令人嘆為觀止。片中飾演這位傳奇飛官的男主角則是由參與過《黑暗戰域》、《外星異種》的性格男星彼得費奧多羅夫，再次獨挑大樑挑戰克服各種地貌氣候的賣力演技。甫在俄國一上映就奪下當週新片票房季軍，票房強壓《惡靈古堡首部曲：拉昆市》、《永恆族》等娛樂鉅片，口碑持續熱映超過一個月。電影一開場的地對空大戰即揭開爆破刺激序幕，跟著主角深入敵軍險境，不到最後一刻絕對無法猜到的震撼結局，是軍事宅、戰爭迷們絕對不容錯過的二戰經典作品，覺得好久沒有感受到大銀幕的轟炸刺激嗎？《逃出封鎖線》將再次引領觀眾體驗視覺感官新風潮！

Precision = 92.3076923076923%  
Recall = 100.0%  
12  
12  
13

In [ ]:

In [ ]: