National Chengchi University

Beat Google

Data Structure - Final Report

January 2022

Group 14

108509315 徐嘉妤

109306032 林怡蕙

109306049 葉於芊

109306064 莊喻絜

1. Topic

Movies

2. Motivation

When we were discussing the topic of the project, we discovered that all of our members have a common interest: movies. However, we encounter a problem that when we search on Google, the result is not always about films. Hence, we want to create a system that prioritizes websites about movies, so it is easier for film lovers like us to find the websites of our interest.

3. Keywords & Weights

Keywords	Weights	Keywords	Weights	Keywords	Weights
電影	3	客串	1	Oscars	1
影展	2	劇情	1	Golden Horse Awards	1
票房	2	故事	1	GHA	1
二輪戲院	2	movie	3	Golden Lion	1
爛番茄	2	film	3	blockbuster	1
大銀幕	2	cinema	2	soundtrack	1
好萊塢	1	theater	2	director	1
奧斯卡	1	the big screen	2	stars	1
金馬獎	1	IMDB	2	cameo	1
金獅獎	1	rotten tomatoes	2	drama	1
鉅座	1	trailer	2	plot	1
演	1	Hollywood	1	acting	1

Keywords in different categories:

Romance					
Keywords	**Weights**	**Keywords**	**Weights**	**Keywords**	**Weights**
浪漫	15	娶	1	lover	1
愛	15	吻	1	couple	1
情	10	慾望	1	relationship	1
喜歡	10	romance	15	marry	1
戀人	1	romantic	15	marriage	1
兩性	1	affection	10	forever	1
婚	1	love	10	desire	1
嫁	1	sexual	1		

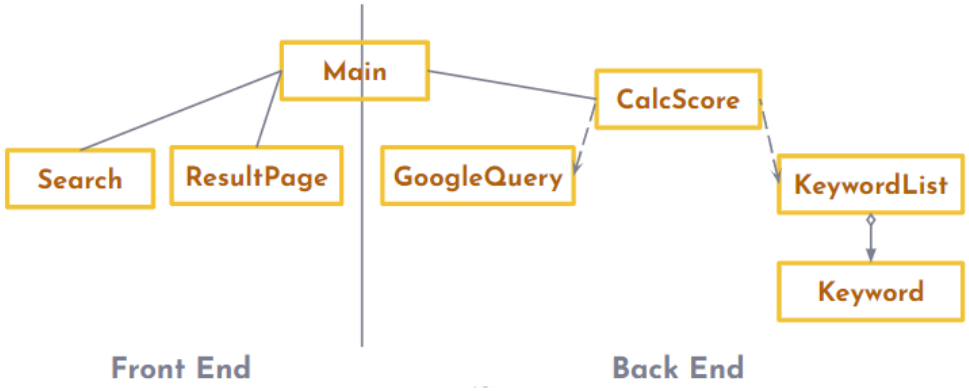
Fantasy					
Keywords	**Weights**	**Keywords**	**Weights**	**Keywords**	**Weights**
奇幻	15	法師	1	time travel	1
科幻	15	巫師	1	robot	1
星際	1	童話	1	monster	1
太空	1	神	1	magic	1
外星	1	鬼	1	power	1
宇宙	1	龍	1	sorcerer	1
地球	1	fantasy	15	imagination	1
時間旅行	1	sci-fi	15	fairy	1
機器	1	science fiction	15	tale	1
妖	1	alien	1	myth	1
怪物	1	space travel	1	god	1
魔	1				

Action					
Keywords	**Weights**	**Keywords**	**Weights**	**Keywords**	**Weights**
動作	15	特技	1	military	1
冒險	15	賽車	1	spy	1
戰爭	1	飛車	1	martial art	1
諜	1	action	15	blast effect	1
武	1	adventure	15	special effect	1
特效	1	war	1		

Comedy					
Keywords	**Weights**	**Keywords**	**Weights**	**Keywords**	**Weights**
喜劇	15	comedy	15	funny	1
笑	1	sitcom	5	hilarious	1
幽默	1	romcom	5	humor	1

Thriller					
Keywords	**Weights**	**Keywords**	**Weights**	**Keywords**	**Weights**
懸疑	15	靈異	1	horror	15
驚悚	15	駭人	1	murder	1
恐怖	15	鬼片	1	detective	1
燒腦	1	顫慄	1	crime	1
推理	1	嚇	1	supernatural	1
謀殺	1	毛骨悚然	1	terrify	1
犯罪	1	thriller	15	horrify	1
殺人	1	suspense	15		

4. Class Diagram



GoogleQuery	fetchContent(String): String fetchGoogle(String): String query(String): HashMap<String, String>
CalcScore	countKeyword(String, String): int calcScore(): HashMap<Integer, ArrayList<String>> sort(HashMap<Integer, ArrayList<String>>): List<Integer>
KeywordList	getKeywordList(): ArrayList<Keyword> filter(String): String movie(): void action(): void romance(); void comedy(): void fantasy(): void thriller(): void

5. Target Users

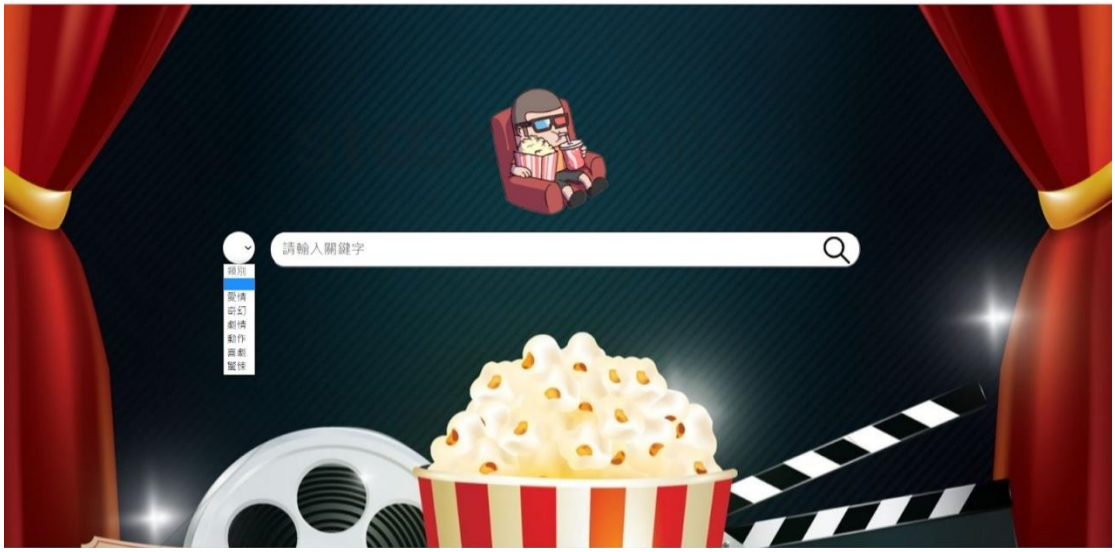
Movie lovers, or simply people who want to search for information related to movies.

6. Purposes of Using the System

We want to filter the results on the Internet and provide information related to movies first, so that users could find these results more easily.

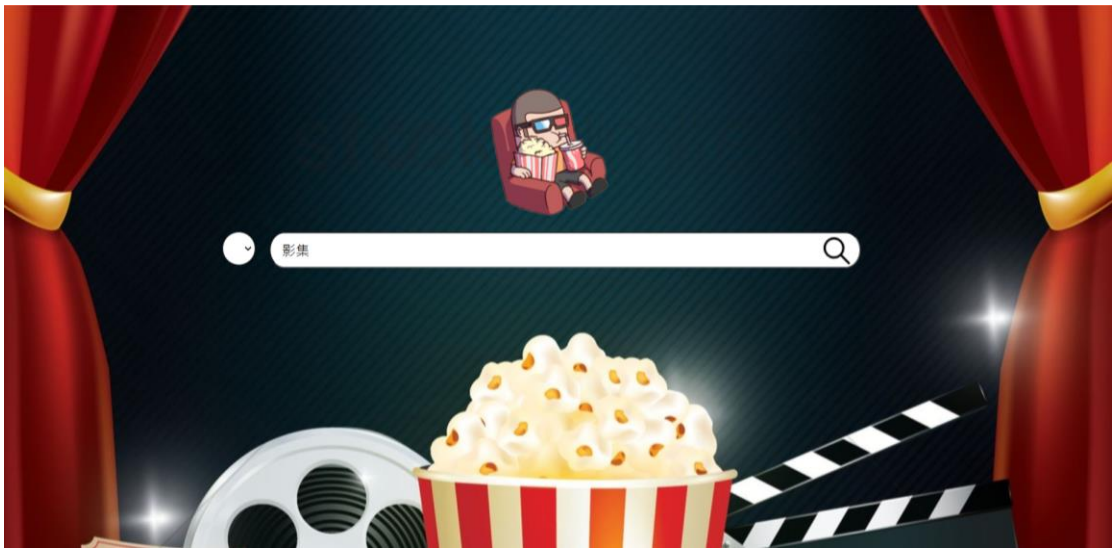
7. System Functions

(1) Several categories to help the system be more accurate; if none of the categories is selected, then the system will just research based on the first set of keywords from 3.

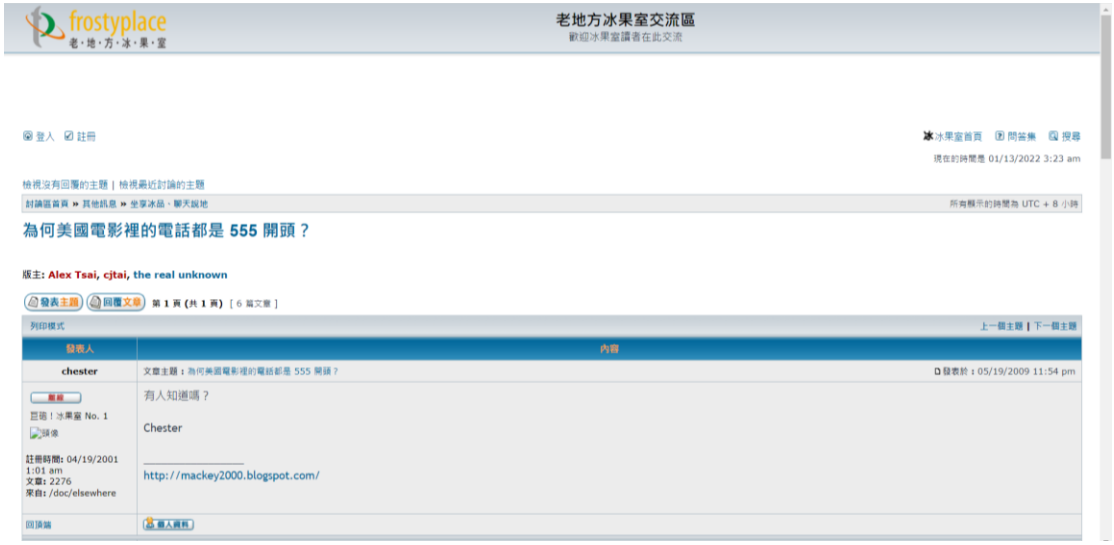


(2) The system automatically calculates the total scores of each website

(3) All the websites are ranked and shown on the result page; the system shows only 30 websites by default



(4) Click on any link and it will lead to the original website



(5) Besides Chinese and English, the system also supports Japanese, and Korean

8. How to Use

(1) Type keywords in the search bar

(2) Choose which category; if not chosen, the system will just show results about movies, regardless of categories

(3) Click the “search” button

(4) Click the icon on the left of the search bar to go back to the main page

9. Test Plans

(1) Walk-through

(2) Desk checking

(3) Unit testing

(4) Integration testing

(5) Performance test

(6) Alpha test

10. Schedule

Project initiation	10/18							
Structure design		10/29~ 11/15						
Back end coding			12/25~ 1/1					
Front end coding				1/2~ 1/10				
Testing & debug					1/6~ 1/10			
Report						1/6~ 1/11		
Presentation preparation							1/12~ 1/13	
Demo								1/13

11. Work Distribution

徐嘉妤	Page rank Site rank	莊喻絜	Web page
林怡蕙	Call Google Related keywords Presentation & Demo	葉於芊	PPT Final report

12. Problems Encountered & Solutions

(1) Don’t know how to connect front end with backend

(2) Don’t know how to convert the final result into a webpage

(3) Results are not necessarily ranked according to the given category

(4) Servlet response 400/403/404 bad request

Solutions:

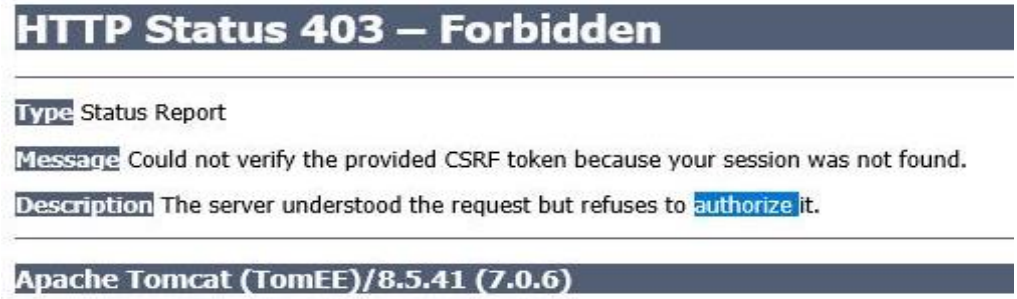
```
// search google
String url = "http://www.google.com/search?q=" + searchKeyword + "&oe=utf8&num=30";
System.out.println("url: " + url);

String content = fetchGoogle(url);
HashMap<String, String> retVal = new HashMap<String, String>();

Document doc = Jsoup.parse(content);
Elements lis = doc.select("div");
lis = lis.select(".kCrYT");

for (Element li : lis) {
    try {
        // parse down URL link
        String citeUrl = li.select("a").attr("href"); // System.out.println("origin: " + citeUrl);
        if (citeUrl.startsWith("/url?q=")) {
            citeUrl = citeUrl.replace("/url?q=", "");
        }
        String[] splittedString = citeUrl.split("&sa=");
        if (splittedString.length > 1) {
            citeUrl = splittedString[0];
        }

        // url decoding from UTF-8
        citeUrl = java.net.URLDecoder.decode(citeUrl, StandardCharsets.UTF_8);
        citeUrl.replaceAll(" ", "%20");
```



Solutions:

```
public String fetchContent(String url) throws IOException {
    String retVal = "";

    try {
        URL u = new URL(url);

        HttpURLConnection.setFollowRedirects(false);
        HttpURLConnection conn = (HttpURLConnection) u.openConnection();
        conn.setRequestProperty("Accept", "*/*");
        conn.setRequestProperty("Connection", "Keep-Alive");
        conn.setRequestProperty("User-Agent",
                "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Chrome/96.0.4664.110 Safari/15.2");
        CookieHandler.setDefault(new CookieManager());

        if (conn.getResponseCode() == 403 || conn.getResponseCode() == 400 || conn.getResponseCode() == 404) {
            retVal = url;
            System.out.printf("Error %d: %s\n", conn.getResponseCode(), retVal);
        } else {
            InputStream in = conn.getInputStream();
            retVal = new String(in.readAllBytes(), StandardCharsets.UTF_8); // will fail with large amounts of data
        }
    } catch (MalformedURLException e) {
        e.printStackTrace();
        retVal = url;
    } catch (IOException e) {
        e.printStackTrace();
        retVal = url;
    }
```

(5) Java.IO.Exception Premature EOF

```
确。java.security.InvalidKeyException: IOException: Detect premature EOF
content={"code":"10000","msg":"Success","out_trade_no":"20191009681975330","qr_
code":"https:\/\/qr.alipay.com\/bax08876xtkegkpv9lxg50d5"}, charset=utf-8, public
Key=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsH3BzT1SOGRniX92HOwlJmhGBaYb7k/3
cqakd6mk6pf919zoKLPb9uh91m7eccP/t9YWzxF3BwvsclXEux/bXrRCSVoHZPKonHvQwwynzAPoRjp
BvZP8/12OOgm/yCB9ytkKDydwxCkc9dHAFuVcsA19v9iGqyS4E28P9cXjWn08Lh4BzP26a2bY0M1GC1
iHPgNTDz1nSdH6eTGIKFJKyI9rPxQefio/nK7H1dlccHcmOuU8LeYDqXBWAwt54JOMQ75Q2LnYV4l+X
EGxTlwjhB/nhdl6oufqb1Cwko5dcmg74rXYCGZQWyLRvADg
```

Solutions:

```
public String fetchContent(String url) throws IOException {
    String retVal = "";

    try {
        URL u = new URL(url);

        HttpURLConnection.setFollowRedirects(false);
        HttpURLConnection conn = (HttpURLConnection) u.openConnection();
        conn.setRequestProperty("Accept", "*/*");
        conn.setRequestProperty("Connection", "Keep-Alive");
        conn.setRequestProperty("User-Agent",
                "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Chrome/96.0.4664.110 Safari/15.2");
        CookieHandler.setDefault(new CookieManager());

        if (conn.getResponseCode() == 403 || conn.getResponseCode() == 400 || conn.getResponseCode() == 404) {
            retVal = url;
            System.out.printf("Error %d: %s\n", conn.getResponseCode(), retVal);
        } else {
            InputStream in = conn.getInputStream();
            retVal = new String(in.readAllBytes(), StandardCharsets.UTF_8); // will fail with large amounts of data
        }
    } catch (MalformedURLException e) {
        e.printStackTrace();
        retVal = url;
    } catch (IOException e) {
        e.printStackTrace();
        retVal = url;
    }
```

(6) Server redirected too many times (20)

```
Exception in thread "main" java.net.ProtocolException: Server redirected too many  times (20)
        at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown Source)
        at sun.net.www.protocol.https.HttpsURLConnectionImpl.getInputStream(Unknown Source)
        at java.net.URL.openStream(Unknown Source)
        at URLReaderWithOptions.main(URLReaderWithOptions.java:58)
```

Solutions:

```
public String fetchContent(String url) throws IOException {
    String retVal = "";

    try {
        URL u = new URL(url);

        HttpURLConnection.setFollowRedirects(false);
        HttpURLConnection conn = (HttpURLConnection) u.openConnection();
        conn.setRequestProperty("Accept", "*/*");
        conn.setRequestProperty("Connection", "Keep-Alive");
        conn.setRequestProperty("User-Agent",
                "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Chrome/96.0.4664.110 Safari/15.2");
        CookieHandler.setDefault(new CookieManager());

        if (conn.getResponseCode() == 403 || conn.getResponseCode() == 400 || conn.getResponseCode() == 404) {
            retVal = url;
            System.out.printf("Error %d: %s\n", conn.getResponseCode(), retVal);
        } else {
            InputStream in = conn.getInputStream();
            retVal = new String(in.readAllBytes(), StandardCharsets.UTF_8); // will fail with large amounts of data
        }
    } catch (MalformedURLException e) {
        e.printStackTrace();
        retVal = url;
    } catch (IOException e) {
        e.printStackTrace();
        retVal = url;
    }
```

13. Expectations for Future Development

(1) Include more keywords to be more accurate

(2) Add more categories to meet the needs of every user

(3) Be able to search for other types besides movies, such as TV series

(4) Support all languages, especially those that are very similar to English and may confuse the system, such as Spanish

14. GitHub Link

https://github.com/109306064/final