Design of Movie Data Visualization System Based on Web Crawler

Nannan Zhang
School of Data Engineering, Pearl River College, Tianjin University of Finance and Economics, Tianjin, 301800, China
*Corresponding author’s e-mail: helk@teda.gov.cn

Abstract. This paper uses Python's requests, beautifulsoup, Jieba and fon tools to obtain the relevant data of all public films in 2015-2020 and analyze them. Finally, the data results are displayed graphically through Ajax technology, layui framework and eckards.

1. Introduction

1.1. Development background and significance of topic selection
The rapid development of informatization now makes the sharing of information on the Internet convenient and fast. Nowadays, the information on the Internet is very complicated. There are a large number of pictures, text, and video resources on the Internet every day. These resources provide people with rich content and make it easier for people to obtain information. But how to obtain the required information and resources from the massive amount of information has become a big problem[1]. Network storage is still being improved, and digital TV-related industries are also developing rapidly, which has also promoted the vigorous development of related film industries. A large number of movie resources appear on the Internet, and user preferences can be judged through user analysis. However, it is difficult to analyze some relevant data of the movie. This is because the type and amount of movie data is very large, and it also contains a lot of unnecessary information, and the form of display is not specific and visual.

1.2. Related research at home and abroad

1.2.1. Current research status of movie data collection
There are currently two mainstream methods for obtaining data. One is that the platform will provide a public application programming interface (API) through which data can be obtained[2]. Although data can be easily and quickly obtained through open APIs, the shortcomings are very obvious, and its open permissions are extremely limited. For example, many data interfaces are not free, and even worse, they cannot be obtained from the outside world. Even if the API is open to the outside world, there are strict controls on the frequency and number of data accesses.

The other is to obtain data through web crawling technology, which is currently used at a very high rate. We can use web crawler technology to crawl relevant web pages to collect the required data. The difficulty of this method is to deal with the anti-crawler mechanism. The advantage of the web crawler is that it is not restricted by the official API, can obtain more comprehensive data, and can collect a large amount of platform data at the same time. The obvious disadvantage of this method is that it
requires a higher level of technical personnel, it is difficult to write programs, and the maintenance and protection costs of this system are relatively high in the later period[3].

1.2.2. Visualization research status
Visualization technology uses computer algorithms to convert some seemingly unrelated data into images for display combined with image processing technology. At present, it has been widely used in decision processing, data analysis and web page visualization[4].

At present, the visualization research system abroad has been approaching maturity. For example, Wanner and others co-designed the visualization system ForAVis. There is also a visualization product SASR Visual BI that is more suitable for daily use, which presents the effect of dynamic interaction.However, in China, the upsurge of visualization research has just started, but many visualization tools have been launched, such as The Network liar map of China developed and researched by 360[5]; in addition, there is also a visualization tool developed by Baidu that can be used for Show chart information and it is currently free. Echarts' open source data is regarded as commercial grade. Using Echarts tool to make charts can be dynamically displayed on the webpage and is convenient for research and analysis of data[6]. Since Echarts's code is open source, it is currently used by people engaged in development for research and development.

2. demand analysis
This system is to crawl the movie website real-time data, clean and integrate the data, store the acquired data resources in the database, visualize the data, and display it through HTML web pages, mainly in the form of images Displaying from different angles allows users to analyze relevant movie data more clearly and intuitively[7].

2.1. System functional requirements
The functional requirements of the system mainly come from the following three aspects:

The first aspect is to collect basic information about movies released by Maoyan Film Network from 2015 to 2020, including movie type, rating, box office, director, etc., to solve the data source problem of the system, and to provide guarantee and support for the subsequent visualization work. The second aspect is to perform multi-angle analysis of movie data to complete the pre-processing of movie data, including data deduplication, field cleaning and correction. Its main job is to analyze the box office in multiple dimensions to satisfy producers and film industry workers. Insights and development forecasts for the film market. In the third aspect, the data analysis of movie information is presented in a visual form.

2.2. System non-functional requirements
In addition to the above-mentioned functional requirements, a certain degree of requirements must be imposed on the reliability and safety of the system. Explained through the following three points:

- The foundation of the visualization system comes from movie data, and the visualization system should obtain data in a timely manner to meet the requirements of users who want the data to be safe, reliable and instant.
- In order to ensure the performance of the system, the visualization system should complete the user's front desk request and display the data within 0.5 seconds to improve the user experience.
- Data storage is required to be safe and private, and data backup processing should be done in time.

3. the overall design of the system

3.1. Brief introduction of development technology
This section introduces some technologies and development environments that will be used in the movie data visualization system. The development techniques used mainly include:Front-end LayUI
The development work of this article is completed under the Windows operating system environment, in which the Python development environment is created by installing Anaconda, and the program is written by pycharm.

3.2. The logical structure of the system
The movie data visualization system uses the front-end and back-end separation mode for system construction. The front-end uses the LayUI framework for web interface development. At the same time, Echarts displays visual graphics. The back-end movie data crawler uses requests and font anti-crawling FontTools tools for data. Crawling, while borrowing the jieba library for word segmentation, and MySQL database for data storage. Establish a relationship between the Flask framework interface and HTML, and Ajax is used for front-end and back-end interaction[8].

3.3. System function structure
The system design mainly includes two parts. Part of the film data collection system uses pycharm to program through the back end of the system, mainly for data collection, sorting, cleaning, and classification, and finally the required data is stored for transmission for visual display. The other part is the visual display system, which is mainly responsible for making the analyzed data into images and displaying them through the web interface.

4. The detailed design and implementation of the system

4.1. Design of main functional modules of the system

4.1.1. Design of function module for collecting movie information
- Collected data source and data type
  The crawler mainly crawls the main information of the movie, including basic information such as the name of the movie, the language of the movie, the cast and crew, and the rating.
- Design of film information collection function
  Python is used for data crawling. Since the target website for crawling is Maoyan Movies.com, it is a website with anti-crawler mechanism and man-machine verification mechanism. At the same time, some text information is encrypted. Therefore, in the process of crawling data, a simulated browser access is used, and time sleep (10) is also used in the program to reduce the frequency of access to imitate human actions. Through response=requests.get(url), the get method of requests is called. Get the returned object, and use Fonttools to parse the encrypted data during the crawling process.

4.1.2. Film visual analysis function module design
The visual analysis function module displays the results of data processing on the user interface with the aid of drawing plug-ins. The main functions of this visualization system include the user login system and the detailed display of movie data. The data overview mainly includes basic information such as movie name, box office, rating, director, and duration. The visual display of movie data includes four modules, box office share, box office list, box office changes, and actor models.

The administrator logs in to the system and obtains the relationship mapping of the database through Flask. It is necessary to map each request and the Python function, and display the data on the page, and then display it to the task object.

4.2. Data visualization realization process

4.2.1. Implementation of Flask framework
First import the Flask extension, create an instance of the Flask application, and pass in __name__ to
determine the path of the resource. The program that maps each request and the Python function, and handles the relationship between the URL and the function is called routing. Defining the route is implemented through the decorator, and finally starts the program to run.

Other data interface processes are basically similar. The system provides a total of 17 interfaces, which are provided to the front-end page to obtain data. For example @app.route('/page2',methods=['GET'])def page2():
code show as below:

```python
def page2():
    if(len(request.args)!=0):
        data_year = request.args['year']
        data_top = request.args['top']
        sql = "SELECT name,box_office from files WHERE year = "+data_year+" ORDER BY box_office DESC LIMIT 10"
    else:
        sql = "SELECT name,box_office from files ORDER BY box_office DESC LIMIT 10"
    conn = pymysql.connect(host='127.0.0.1', user='root', password='123456', port = 3306, charset='utf8mb4')
```

The Jsonify function is for users to process the returned serialized json data (serialized json object). To use Jsonify, you need to add a sentence app.config['JSON_AS_ASCII'] = False. This is mainly to avoid displaying Chinese garbled characters. code show as below:

```python
app.config['JSON_AS_ASCII'] = False
CORS(app, supports_credentials=True)
```

### 4.2.2 LayUI framework

The front-end and back-end transfer data through JSON, and through the advantages of layui's componentization and modularization, it reduces code duplication and improves development efficiency. The whole page is divided into template and script.style. The template tag stores the html code to realize the page layout, and combines with the style code to set the page effect. This article uses a lot of LayUI framework programming code, so I won't go into details one by one. Part of the code that realizes the function is displayed. Show the programming code of the HTML interface header and sidebar respectively. code show as below:

```html
<div class="layui-header">
   <div class="layui-side">
      <img src="/assets/images/logo.png"/>
   </div>
</div>

4.2.3 Ajax technology implementation

The front-end module sends out an HTTP request through Ajax, and the data interface responds to the request and returns the data to the front-end JS object. Ajax loads remote data via HTTP requests. The programming used in this article is that $.ajax() returns the XMLHttpRequest object created by it. The callback function after the request is successful. The data returned by the server and processed according to the dataType parameter; a string describing the state.
code show as below:

```javascript
function init() {
    console.log('initialization.');
    $.ajax({
        type: 'GET',
        url: 'http://127.0.0.1:5000/page2',
        success: function(msg) {
            mychart.setOption({
                tooltip: {
                    show: true
                },
                series: [{
                    type: 'wordCloud',
                },
            ]
        }
    });
}
```

Since the visualization module implements the word cloud display form, the type is "wordCloud".

### 4.2.4. ECharts implementation

Using the EasyWeb background development framework, the foreground LayUI binds the background data through the Ajax interface. Write the DOM container to display the chart presentation data; initialize the ECharts instance and bind the corresponding container; specify the parameter configuration items of the chart, such as the title, color, style and other information of the chart; bind the corresponding data. The statistical chart on the page is implemented by ECharts, and the code for the part of ECharts drawing chart is as follows:

```javascript
series: [{
    type: 'wordCloud',
    gridSize: 6,
    shape: 'diamond',
    sizeRange: [12, 50],
    width: 800,
    height: 500,
    textStyle: {
        color: function() {
            return 'rgb(' + (Math.random() * 160),
        }
    },
}
```

Since the implementation of ECharts for visualizing the four pages is basically similar, the following discussion will not be repeated.

### 4.3. Data visualization realization results

The two parts of the data visualization results in this article are divided into six pages for display. The first part is the home page and data overview, and the second part is the visualization page display. It is divided into four pages: box office list, box office proportion, box office changes, and actor model workers. Display movie data from all angles.

The landing page is divided into two pages: administrator and user, which are used to meet the different needs of different personas. The lower part shows the user login completion page, and the personas in the upper right corner is the user. As shown in Figure 1.
4.3.1. Data overview display
This page displays the crawled and processed data on the front end in the form of data visualization. The data is comprehensive and clear, and the main information of the movie can be intuitively understood from multiple angles. Each page displays ten data items, a total of 75 pages, 750 data items.

4.3.2. Four visualization modules
The box office percentage visualization module displays the results of different types of box office statistics and their comparisons. The box office percentages of all types of movies are displayed through two forms of histogram and rose chart. You can intuitively see the box office numbers of different types of movies. The search bar can search the box office percentage of the corresponding year and month. As shown in Figure 3.
The box office list visualization module is displayed according to the number of box office in ascending order. By default, all the top 20 movies in the box office are displayed, displayed in the form of a word cloud, and the box office list of the corresponding year and month can be searched according to the search bar. As shown in Figure 4.

![Figure 3. Visual display of the box office list](image)

The box office change visualization module is intended to show the box office changes of different types in different years. By default, it displays the change trend of all box office numbers from 2015 to 2020. You can search for the required type of box office changes according to the search bar, and display it in the form of a line chart as shown in Figure 5. Shown.

![Figure 4. Visual display of box office changes](image)

Figure 4 shows the actor model worker visualization module. This module displays the ranking of the number of movies played by the actors. By default, the top ten actors in all films appearing in the film are displayed, which is called the actor model worker. It is displayed in the form of a histogram and a word cloud. For comparison and display, you can search for actors and model workers of the corresponding year according to the search bar. As shown in Figure 6.
Figure 5. Visual display of actor model worker

5. system test

The test of the movie data visualization system is mainly functional test. Through system testing, it is judged whether the system can meet the needs of users, including the display of interface information, interface jump, verification of information format, and whether the realization of system functions is reasonable. The visualization system in this article is mainly used to display movie data, including movie type, rating, box office, director and many other aspects. The test results of this system are mainly displayed in the form of tables.

Film data collection function test example is shown in Table 1.

| Test item                  | Test step                              | Expected result                                      | Test result              |
|----------------------------|----------------------------------------|------------------------------------------------------|--------------------------|
| The acquisition of movie information | Start the crawler of Maoyan Movie       | generate relevant data of Maoyan Movie platform in the database | in line with expectations |

The analysis function test case of movie data is shown in Table 2.

| Test items                  | test step                              | expected results                                      | test results              |
|----------------------------|----------------------------------------|------------------------------------------------------|--------------------------|
| Box office changes over time | click the year and month search button | a line chart of changes appears on the page           | in line with expectations |
| Percentage of box office    | click the year and month search button | a bar graph of changes appears on the page            | which is in line with expectations |
| The number of actors        | click the year and month search button | a bar graph of changes appears on the page            | which is in line with expectations |
| shooting changes over time  |                                        |                                                      |                          |
6. summary and outlook

6.1. Summary
This paper designs and implements a movie data visualization system based on the Maoyan movie platform. It mainly displays the main basic information of the movie, focusing on the multi-angle display of the movie box office and the data analysis of the relationship between actors and movies. The movie data is crawled through the python web crawler, which uses the requests library and Fonttools tools to deal with the anti-crawler mechanism and text encryption of the cat's eye movie. Use MySQL database to store processed data, use Flask Web development framework to write front-end interface and then obtain data through Ajax asynchronous technology. The LayUI framework is used in the front-end for the development of the Web interface, and the Echarts plug-in is used to display the results of data analysis to users in the form of visual images.

6.2. Outlook
The visualization system designed in this article still has many shortcomings, such as the failure to realize the real-time update of background movie data, and the failure to add emotional analysis processing of film reviews. In the future, it is hoped that the existing system functions can be more comprehensively increased in these two aspects. supplement.

References
[1] Cuixia Mu. Visual design of commodity evaluation text based on Python and echarts [J]. Computer knowledge and technology. 2020,16 (35): 11-14
[2] Zhi Wang. Design and implementation of data mining visualization system for movie website [D]. Huazhong University of science and technology, 2018
[3] Haoshi Yu, Kuang Fangjun. Analysis and application of anti crawler technology based on Python [J]. Intelligent computer and application, 2018, 8 (04): 112-115
[4] Junxiong Sun. Research on Web information collection technology based on web crawler [D]. Dalian Maritime University, 2014
[5] Lili Pei. Crawler and word cloud production of movie review data based on Python language [J]. Information recording materials. 2020,21 (05): 116-118
[6] Feng S, Song K, Wang D, et al. A word-emoticon mutual reinforcement ranking
[7] Wide Web-internet & Web Information Systems, 2015, 18(4): 949-967
[8] Wigginton B, Gartner C, Rowlands I J. Is It Safe to Vape? Analyzing Online Forums Discussing E-Cigarette Use during Pregnancy. Women s Health Issues, 2016, 27(1): 93-93