**JD Comments Spider**

1245936 Hermann

1245926 Howard

1245983 Carlos

1245957 Eric

**Abstract**

With the rapid development of the Internet, the World Wide Web has become the carrier of a large amount of information.Search engines, such as the traditional generic Search Engine AltaVista, Yahoo!Google and others, as a tool to assist people to retrieve information, become users access to the World Wide Web portal and guide.However, these universal search engines also have certain limitation.

In order to solve the above problems, the focused crawler of directional grasping related web page resources arises at the historic moment.

The purpose of the crawler produced by this program is to crawl the comments from the jingdong commodity page to the local for data analysis.

1. **Introduction**

During the 618 Shopping festival, we need to browse a large number of products and their reviews on E-commerce platforms like Amazon and JD.com.

However, Most of popular products' comments often have the following problems:

1. The number of comments is huge and it's impossible to find the information you want to refer to.

2) A lot of meaningless information.

3) Bad reviews that has nothing to do with the quality of the product.(Logistics, customer service)

In order to help seller improve their products and selling strategy, or make reasonable reference for customers, further analysis on comments data should be executed.

Thus, the purpose of this project is: Collect reviews of a certain product on JD.com, and classify them according to good, medium and bad reviews, and write them into EXCEL file ‘CommentData.xls’ for further analysis.

**2.Design principle**

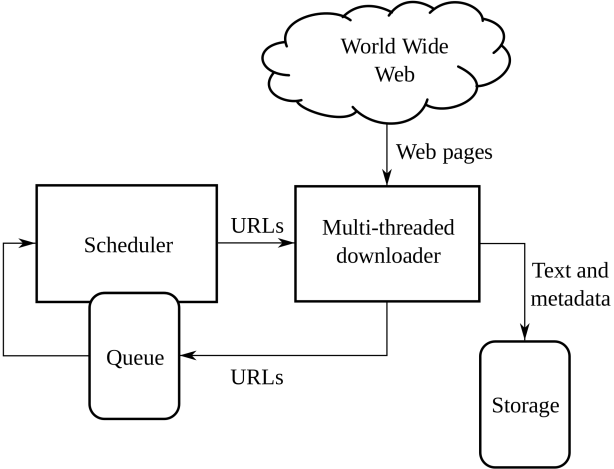
**2.1 The concept of Spider**

Web Spider, or Web Crawler, is a program or script that automatically extracts information from the World Wide Web according to certain rules. The Internet itself is a big web, and the spider is crawling on the web spider, if it meets resources you want, then the spider will grab it down. Spiders will follow the hyperlink to grab information on all of the pages they can.

**2.2 Principle of Spider**

**The procedures of getting data from the server site is consist of three steps:**

1. **Send request message**
2. **Get response message**
3. **Parse pages**

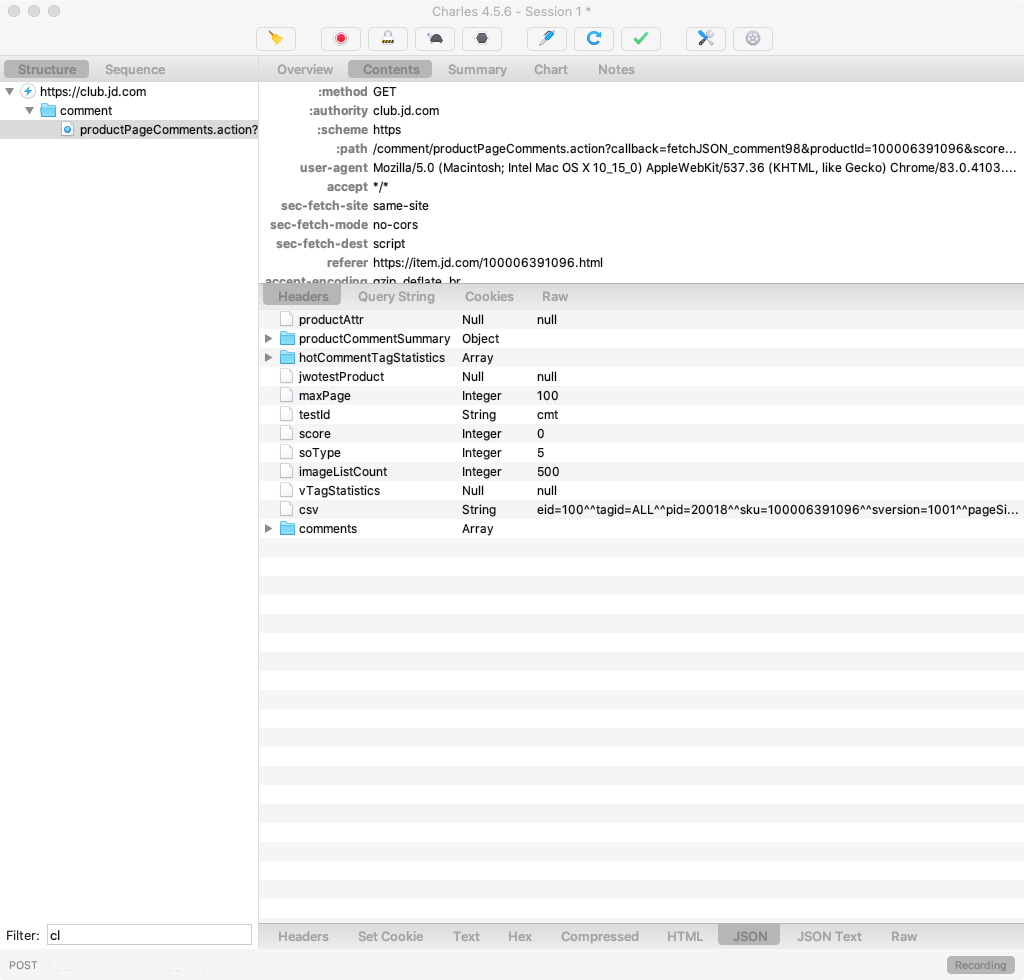


**3.Implementation process:**

The experiment is executed with python 3.8.

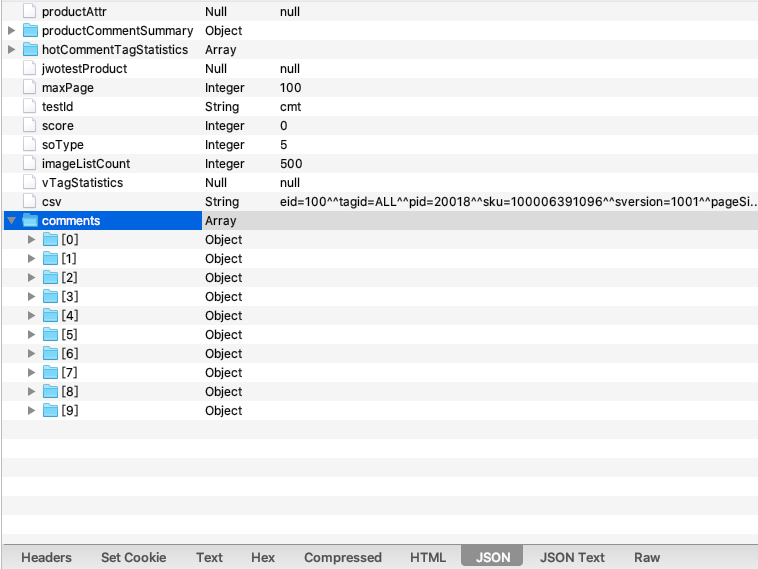
1)Open Google's browser and enter the purchase page of a certain product on JD.com.

2)Use the capturing tool (such as Charles), open the agent and refresh the page to find the API interface of the comment content.



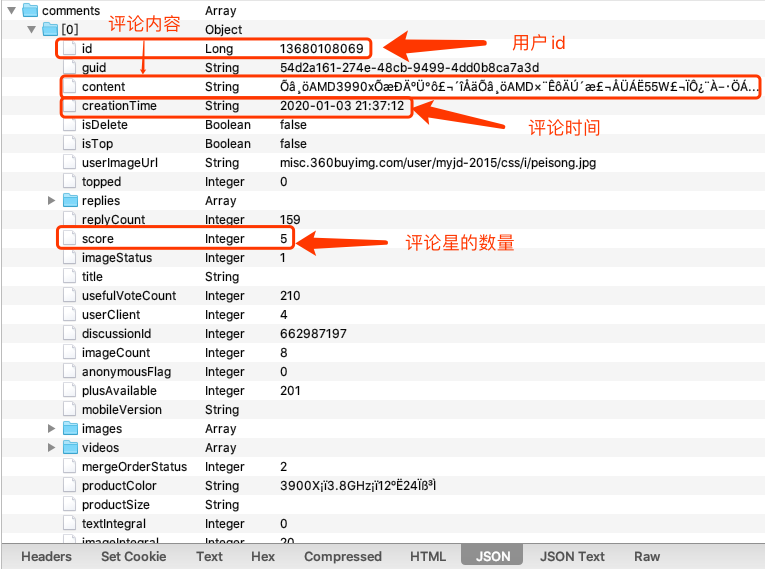
3) Find the product ID and its comment url:

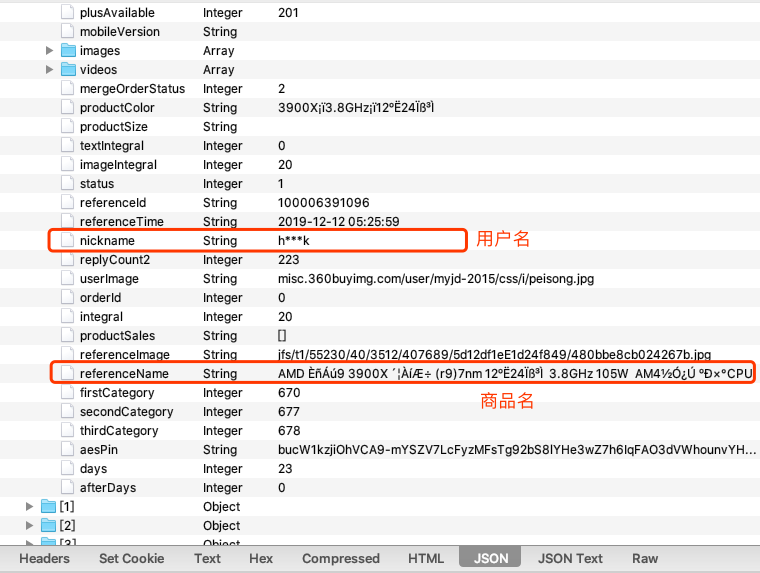


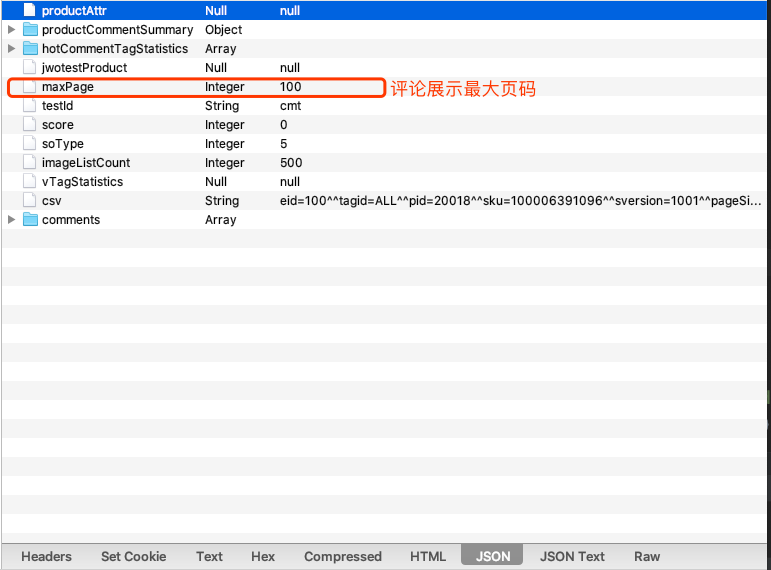


4)Through browser debugging, we can know the meaning of parameters in the comment URL:







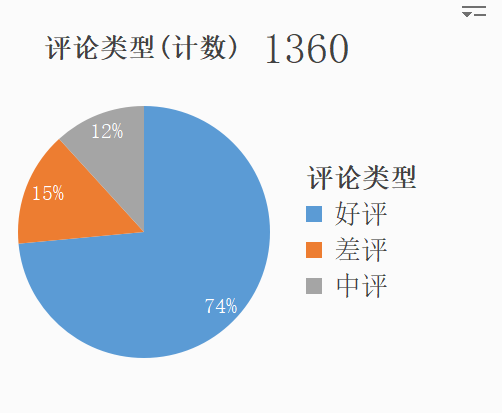
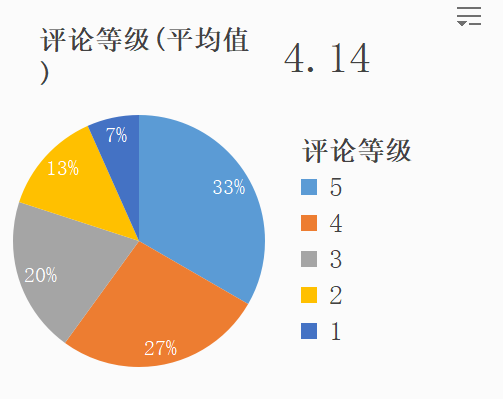


5)After we get the url we want, we could request the comment data and restore them into the excel file.

**4.Conclusion and Data Analysis**

**4.1 Data Analysis**

Through an analysis of 1630 recent reviews, I found that the rate of poor and medium reviews for this product is much higher than the official website shows



Based on those data analysis, customers could make a more considerable decision and sellers could make a promotion of their products more precisely.

**4.2 Conclusion**

By sending a large number of requests, we get a large amount of specific information from the server in a short time and store it locally.

Through spider program, we could collect the information we want more efficiently.

Which help us get meaningful information in an age of complex information.

**4.3 Issues**

1) The amount of data that can be crawled is limited, with only 100 pages of comments.

2) The latest comments cannot be imported in real time.

3) Only a single web site can be accessed. Multiple urls cannot be resolved.

**4.4 Future Works**

1) Add a real-time crawler function for the crawler.

1. Create a text-sentiment analysis program to rate how positive buyers are about the product.

3)Improve the efficiency of keyword fetching.

**Reference**