International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 4 190 - 193

Extracting Reviews of Mobile Applications from Google Play Store

Palak Thakur
ADIT
A.D. Patel Institute of Technology, Gujarat, India

Asmita Raut
ADIT
A.D. Patel Institute of Technology, Gujarat, India

Salina Bharthu
ADIT
A.D. Patel Institute of Technology, Gujarat, India

Prof.Bhagirath Prajapati
Assoc. Professor, ADIT
A.D. Patel Institute of Technology, Gujarat, India

Prof. Priyanka Puvar
Asst. Professor, ADIT
A.D. Patel Institute of Technology, Gujarat, India

Abstract-Nowadays, there are huge numbers of mobile applications coming in the market. Many of these applications provide similar functionality and are available from different unknown vendors. Users are not aware about the risks and malfunctions related to the applications and hence end up downloading an inappropriate application. Thus classification of the applications has become an important aspect. This classification can be done with the help of extracting some data and doing analysis on that data. Nowadays reviews have become an asset for decision making purpose for the users using these mobile applications. Thus classification of the application can be done on the basis of analysis of the reviews. In this paper we are extracting the reviews for various mobile applications using crawling and scraping process and also the use of Json and Jsoup has been discussed.

Keywords- Scraper, Web Crawler, focused crawling, reviews, mobile applications,JSoup, JSon

*****

I. INTRODUCTION

In today's world, the use of mobile device is not limited to only calling and texting. With the passage of time, this device has started providing various facilities among which the most amazing facility is internet service through which the user is able to use email service, access the social networking sites etc. Also due to internet service available in the mobile different types of other applications are also available to the users for the use. These applications have different purposes. Initially there were only few applications available, so the users had not much to think before using the application. But nowadays applications with same purpose are available in ample amount. So the users are really not sure about which application is preferable and end up downloading an inappropriate application. Thus to help the user in selecting the most preferable application, we will provide a risk score to the application on the basis of the reviews for the application. We will be extracting the reviews from the Google Play Store.A Web Crawler is a special program that visits Web site, reads the page and then collects raw contents in order to create entries for further work[1]. Crawler will be used to extract the entire page from Google Play Store. Then we will be focusing on extracting only the reviews from the pages available for a particular application. This will be done through the scraping process. Web scraping is a technique of extracting useful information from a specific website. Its main purpose is to extract the required information and then transform it into a format that can be useful in some another context. We will be focusing only on the textual content of the reviews and not other information like date on which the review was written, the rating given by the user, the title given by the user for the particular review etc. Then finally to save the extracted reviews in some useful format, we will be using the Json and Jsoup.Jsoup is an open source java library consisting of methods design to extract and manipulate html document content. Json is used to list all elements that you want to crawl.

II. WEB CRAWLER FOR EXTRACTING PAGES

A Web Crawler is a special program that visits Web site, reads the page and then collects raw contents in order to create entries for further work[1]. It is also known as a "spider" or a "bot"[1]. The behaviour of Web Crawler is the result of a combination of policies[3]:

- A selection policy which states the pages to download.
- A re-visit policy which states when to check for changes to the page.
- A politeness policy that states how to avoid overloading websites.
- A parallelization policy which states how to coordinate distributed web crawlers.

Here is a process that web crawler follows for extracting the pages from Google Play store[1]:

- Starts from one preselected page which is the page indexed by the URL for that Mobile application. We call the starting page the "seed" page.
- It then extract all the links available on that page and stores them in the URL queue.
- Then it follows each of those links to find new pages for user reviews or other application details.
- Extract all the links from all the new pages found.
- Follow each of the links to find new pages.
- And the procedure then continues till it keeps on finding new links or else some termination condition is fulfilled.

III. FOCUSED CRAWLING

A focused crawler is web crawler that attempts to download only those web pages that are relevant to a pre-defined topic or set of topic[4]. A focused crawler is called the topical crawler because fetch only those pages that are topic specific[4].

A focused crawler tries to get the most promising links, and ignore the off- topic document. A focused crawler must predict the probability that whether an unvisited page will be relevant before actually downloading the page.

A possible predictor is the anchor text of links.The performance of a focused crawler depends on the richness of links in the specific topic that is being searched, and focused crawling usually relies on general web search engine for providing starting points[7].

A. Architecture of focused web crawler

The Architecture of the focused Web crawling is shown here in figure 1, in the architecture URL Queue contains the seed URLs maintained by the crawler and is initialized with unvisited URLs. Web page Downloader fetches URLs from the URL Queue and Downloads corresponding pages from the internet. The parser and extractor extract information like the text and the hyperlink URLs from a Downloaded page[4].

Relevance calculator calculates relevance of a page with respect to topic and assigns score to URLs extracted from the page. Topic filter analyses whether the content of parsed pages is related to topic or not. If the page is relevant, the URLs extracted from it will be added to the URL queue, otherwise added to the irrelevant table[4].

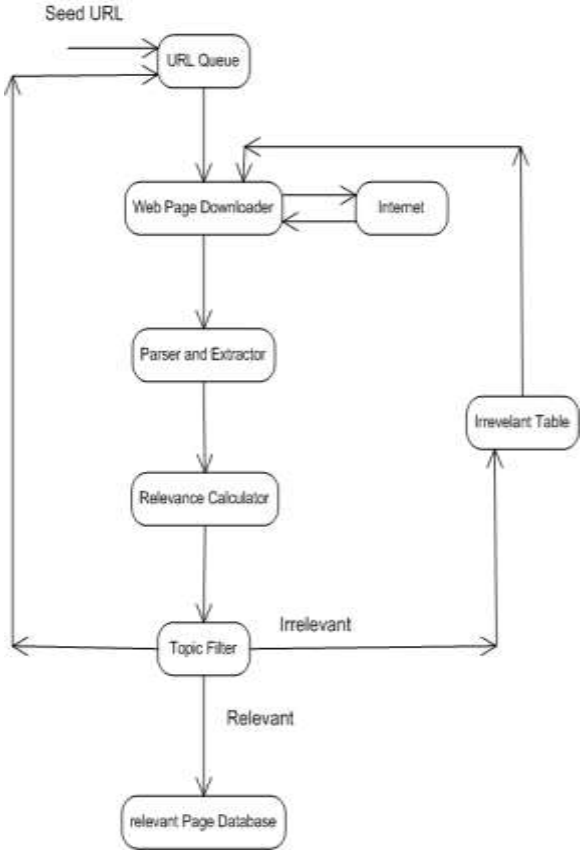

Fig. 1 Architecture of Focused Crawler

IV. WEB SCRAPER FOR EXTRACTING REVIEWS OF MOBILE APPLICATIONS

The term scraping refers to the process of parsing the HTML source code of a web page in order to extract or scrape some specific data from the web page[2]. The target web page does not need to be opened in a browser. Only an internet connection is required[2]. But the target web page needs to be public (not behind a login or encrypted)[2]. The basic process of scraping can be easily understood as shown in figure 2.

The whole process takes place in the form of request and response. Initially we need to mention the URL from which the data has to be extracted. Then an object of URL class is created which uses the URL mentioned for scraping the data[6].Then a call to the openConnection method is made using this object. The openConnection method returns aURLconnection object, which is used to create a BufferedReader object. The BufferedReader objects are used to read streams of text, probably one line at a time[6].

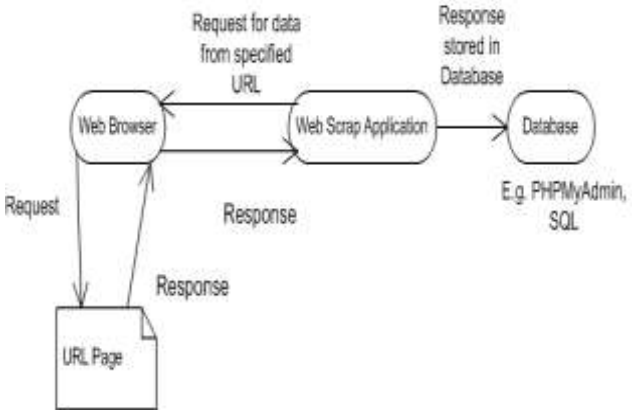

Fig. 2 Scraping Process

But also at the same time it is important to set a limit for the data to be retrieved from the URL as mentioned. So we can set a buffer size too[6]. Now to save the data extracted in a well formatted document the use of JSON and JSOUP is done which has been discussed in the next section.

Thus for our work we require the reviews of different applications. Therefore here the process of scraping is applied where the URL to be mentioned will be the complete URL to reach the page of a particular application. Here we are using the Application Id of a particular application. Then using the scraping process mention in the section, the reviews for a particular application will be scraped.

In the below given code snippet, an object con of class MyConnection is created. Then we created a variable of type String which holds the URL of the page that needs to be scraped. To provide a terminating condition for a scraper we have used the integer variable MAX_REVIEW_COUNT whose value is 2000. Now again we have another variable of type String which holds the application id of the particular application. Now from the main function a constructor of ReviewExtract class will be called in which the application id

will be passed as a parameter. This constructor will create a new XML file at the mentioned location. Now a http request will be send and control will be transferred to the sendpost() method which is called using the object of ReviewExtract class.

```
MyConnection con=new MyConnection();

private static final String url =
"https://play.google.com/store/getreviews";

private static final int MAX_REVIEW_COUNT = 2000;

private static final String INSTAGRAM =
"com.instagram.android";

publicReviewExtract(String applicationId){

this.applicationId = applicationId;

file = new File("d:/"+applicationId+".xml");}

public static void main(String[] args) {

ReviewExtract http = new ReviewExtract(INSTAGRAM);


try{System.out.println("Send Http POST request");

file.createNewFile();

try (FileWriterfw = new FileWriter(file.getAbsoluteFile())) {

fw.write("<app id="+http.applicationId+"\">\n");

 }

while(!end){

http.sendPost();

            }

}

}
```

Now in sendPost() method, an object named obj is created for the URL class. Then using the object obj the openConnection method of HttpsURLConnection class is called. Thus the connection is established to extract the reviews. Then a post request is sent to the mentioned URL. Now using the bufferedReader object the response is taken and then stored in the response object.

```
private void sendPost() throws Exception {

URL obj = new URL(url);

HttpsURLConnection con = (HttpsURLConnection)
obj.openConnection();

try (BufferedReader in = new BufferedReader(
  new InputStreamReader(con.getInputStream()))) {

String inputLine;

StringBuilder response = new StringBuilder();

in.readLine();

in.readLine();

while ((inputLine = in.readLine()) != null) {

response.append(inputLine);

}}
```

V. USE OF JSOUP AND JSON IN OUR SYSTEM

JSON is abbreviation for JavaScript Object Notation, and it is a way to store the information in an organized and easy-to-access manner. In a nutshell, it provides us a human-readable collection of data which we can access in a really logical manner. JSON is a lightweight format that is used for data exchanging. JSON is based on a subset of JavaScript language (way of building objects in JavaScript).Compared to static pages, scraping the pages rendered from JSON is often easier: you simply have to load the JSON string and iterate through each object, extracting the relevant key/value pairs as you go. In our project we use json to list all elements that we want to crawl and also JSON parser to convert a JSON text to a JavaScript object. A JSON parser will recognize only JSON text and will not compile scripts. JSON parsers are also faster.
Once the web page is loaded, there arises the need to sniff around the web page to get specific information. JSoup is an open-source Java library consisting of methods designed to extract and manipulate HTML document content[5]. Jsoup scrape and parse HTML from a URL, find and extract data, using DOM traversal or CSS selectors[5]. JSoup is an open source project distributed under the MIT licence. Moreover, JSoup is compliant to the latest version of HTML5 and parses a document the same way a modern browser does. The main classes of JSoupapi are JSoup, Document and Element.

There are 6 packages in JSoupapi providing classes and interfaces for developing JSoup application.

org.jsoup

org.jsoup.helper(utilities for handling data.)

org.jsoup.nodes(contains HTML document structure nodes)

org.jsoup.parser(Contains the HTML parser, tag specifications, and HTML tokeniser.)

org.jsoup.safety(Contains the jsoup HTML cleaner, and the whitelist definitions)

org.jsoup.select(contains packages to support the CSS-style element selector.)

To fetch the page using jsoup:

1. First you create the Connection to the resource.
2. Then you call the get() function to retrieve the page content
   (Document doc =
   Jsoup.connect("http://google.com").get();)

In our project we adopt JSoup component to extract information and URLs from a web page. The file jsoup-1.7.3.jar can be downloaded on its company website. We need to import it into netbeans. While scraping reviews from Google playstore website, these reviews are located between tags, thus JSoup parser object will first locate "div" tag and then using DOM getter method it will locate tag having class="single-review".

importorg.jsoup.nodes.Document;

importorg.jsoup.nodes.Element;

importorg.jsoup.select.Elements;

Document doc =
Jsoup.parseBodyFragment(body);//System.out.println(doc.html());

Elements reviews = doc.getElementsByClass("single-review");

The parseBodyFragment method will create an empty shell document, and insert the parsed HTML into the body element.

Elements provide a range of DOM-like methods to find elements, and then extract and manipulate their data. The DOM getters are contextual: called on a parent Document they will find matching elements under the document; called on a child element they will find elements under that child. In this way we can extract the data we want.

VI. CONCLUSION

This paper provides insight for extracting selective content from the website. There are two major technologies used for carrying out the selection and extraction are crawling and scraping. For our work we require the reviews of different mobile applications. Therefore here the process of scraping is applied where the URL to be mentioned will be the complete URL to reach the page of a particular application. Here we are using the Application Id of a particular application. Then using the scraping process mention in the section, the reviews for a particular application will be scraped.

VII. FUTURE WORK

Now for our future work, we will be learning about various algorithms used in analysing the reviews scraped and then providing a proper classification of the applications. We will be using one of the machine learning algorithms which will be best for our work. A summary of the reviews will be provided in the form of risk score of the application.

ACKNOWLEDGMENTS

This research paper was supported by the Project Guide Prof. BhagirathPrajapati as well as the institution A. D. Patel Institute of Technology, Gujarat, India.

REFERENCES

[1] An Approach to build a Web Crawler using Clustering based K-Means Algorithm, Nilesh Jain, Priyanka Mangal and Dr. Ashok Bhansali

[2] Five easy steps for scraping data from web pages by Jon Starkweather

[3] Web Crawler, Shalini Sharma, Dept. Of Computer Science, Shoolini University, Solan(H.P), India

[4] Efficient Focused Web Crawling Approach for Search Engine, Ayar Pranav and Sandip Chauhan, Computer and Science Engineering, Kalol Institute of Technology and research centre, Kalol, Gujarat,India.

[5] Research of a professional search engine system based on Lucene and heritrix, Ying hong and Chao Lv

[6] Creating sample web scraper-www.packtpub.com

[7] Focused Crawling, Wikipedia