DEVELOPMENT OF WEB APPLICATIONS CRAWLER DATA COLLECTION FOR MONITORING SYSTEM OF SOCIAL MOOD SOCIETY

Utemuratov Yerik

International Information Technologies University, Kazakhstan, Almaty

Annotation. The purpose of this publication is to develop a “crawler” for a system for monitoring social attitudes of society. The publication describes the crawler development steps, selected algorithms and technologies for development, shows the first experimental results of the "crawler." In the course of research and experiments, the system will be improved and the results will be published in the following articles.

Keywords: sentimental analysis, crawler, dataset, parsing, web sockets, cache, content

Introduction Public opinion today is an important indicator of the state of the socio-economic system, since it reflects the level of social tension. Accounting and control of this level allows you to build strategic planning to ensure sustainable development of the socio-economic system, and in companies, in government or in the state as a whole. In this regard, the monitoring of public opinion is an important and relevant management tool actively used by socio-political, financial, economic and social structures.

Public opinion today is an important indicator of the state of the socio-economic system, since it reflects the level of social tension. Accounting and control of this level allows you to build strategic planning to ensure sustainable development of the socio-economic system, and in companies, in government or in the state as a whole. In this regard, the monitoring of public opinion is an important and relevant management tool actively used by socio-political, financial, economic and social structures.

Billions of posts left by users every month cannot be processed manually during public opinion polls. This fact highlights the need for methods of automated mining of textual information, allowing for a short time to process large amounts of data and understand the meaning of user messages. Understanding the meaning of messages is the most important and complex element of automated processing.

In connection with the above, our research is aimed at analyzing social networks and creating methodological foundations and tools for processing and predicting big data. And also, the study aims to summarize and compare information, based on its collection from both social networks and independent sources of the Republic of Kazakhstan.

For this, we identified the following tasks:

- Creating a tool for collecting data from social networks;
- Definition and development of methods for splitting data into categories. Collection of necessary statistical information on the analyzed social attitudes from certain (for the experiment) independent sources;
- Development of methods and models for analyzing information obtained from independent sources;

- Identification of randomness in the collected information, their evaluation and visualization. Development of evaluation criteria.

Data collection tool

To implement the experimental data, a web-based «crawler» data collection application was developed for collecting text and processing data from WEB information portals. For this, our first action was to create a “DataSet”, so that machine learning algorithms could conduct semantic analysis of the text in certain input data.

As information resources for the experiment, we selected the most popular and most frequently visited portals like www. tengrinew.kz, www. nur.kz and www. zakon.kz. For each portal, a «crawler» (crowler-tracked tractor) was developed to upload news content to the «DataSet». In order for the developed “crawlers” to work for each portal, we performed the following steps:

Step 1: study the HTML content and layout architecture, the principle of downloading news for each portal;

Step 2: for each portal, write your own algorithm for loading and parsing data in the correct order;

Step 3: disassemble the function of the system of cleaning unnecessary words "garbage" from the texts downloaded from the portal to improve the efficiency and accuracy of the machine learning algorithm;

Step 4: optimize the speed of loading and processing data using the SQL language;

Step 5: Build a user-friendly interface for viewing, displaying statistics, and managing data after loading.

Step 6: test, fix the processing speed.

To begin with, we have considered the options for processing and collecting data from the content of information portals. Today, there are various ways to collect data. For example, there are technologies for receiving data directly from the page of downloaded news, it is data retrieval at the level of HTML layout. It all depends on the architecture of the site, and how they load the news. Many modern portals load data using JavaScript technology and the jQuery library. These technologies and applications work in real time. They do not generate HTML layout when the page loads, but load them in the process when the user starts scrolling through the news or clicking on the "next" button.

In such situations, there are two ways to solve this problem.

The first way is to create a “bot” which, as a person, will produce an imitation or simulation of scrolling news so that the “crawler” can pull out the data from there. This method has its advantages, as it will always work regardless of changes in the site layout architecture. But, and there is a big minus, in that it is a very long loading time.

The second way is to study the data loading architecture yourself, and find a JavaScript function that loads the news from a specific site controller.

We have chosen the second option, if we define a controller, then it becomes possible to load all the news in a short time. After that, we studied the ways to download all the news of the selected portals tengrinews.kz, nur.kz and zakon.kz. Among the selected portals nur.kz, it turned out to be the site that runs on JavaScript, and the other tengrinews.kz and zakon.kz display all their content at the HTML level of the layout. Therefore, we had to use the Google Chrome browser console in order to view all HTTP requests and responses, to see their header and URL addresses where they go. Finding the nur.kz news download controller was easy. If you look at the download stream through JavaScript, you can see which link the AJAX request refers to, what parameters it transmits, and how this link returns. In our situation, this link looked as follows:

https://data.nur.kz/posts?search[top_status]=1,2&search[section_id]=1&search[language]=ru&per-page=30&search[status]=3&sort=-published_at&thumbnail=r305x185&_format=json&fields=id,slug,catchy_title,description,published_at,thumb,comment_count,section_id&page=1

Result of JSON format: {"id": 1261958, "slug": "segodnya-v-kazakhstane-otmechayut-den-tr", "catchy_title": "Labor Day is celebrated in Kazakhstan today", "description": "The purpose of the holiday is raise the credibility of the working person", "section_id": 1, " thumb ":" https://i.onthe.io/pogudx6so30dr6asi.r305x185.a4902dec.jpg ", " comment_count ": 66," published_at ":" 2018-09 -25 17:51:04 "}

Thus, we can “pull out” all text data from all portals automatically.

Our next task was to develop an algorithm, select the necessary technologies for downloading content from the portal. In this regard, JAVA was chosen for the main language of the technology, since it is widely used, and it is possible to develop high-speed Web applications with a huge load capacity. For parsing data from web portals, there is a library JSOUP. This library is widely used to manage DOM (Document Object Model) Web page objects. This means using this library you can download content from the page. This library is widely used in the manipulation of HTML content and page layouts, or to generate any page. For data storage, the database system MYSQL was chosen, which is capable of storing a huge number of words. The parsing algorithms for all the portals are identical, but the method of manipulating DOM objects is different for everyone. After all, each portal has its own design and its own HTML code. For example, at the portal “nur.kz” the content of the news is displayed in the tag. This is to say, the system pulls out all the text content from this tag. Portals www.nur.kz and www.zakon.kz , and portal tengrinews.kz - .

Our next task was to develop an algorithm, select the necessary technologies for downloading content from the portal. In this regard, JAVA was chosen for the main language of the technology, since it is widely used, and it is possible to develop high-speed Web applications with a huge load capacity. For parsing data from web portals, there is a library JSOUP. This library is widely used to manage DOM (Document Object Model) Web page objects. This means using this library you can download content from the page. This library is widely used in the manipulation of HTML content and page layouts, or to generate any page. For data storage, the database system MYSQL was chosen, which is capable of storing a huge number of words. The parsing algorithms for all the portals are identical, but the method of manipulating DOM objects is different for everyone. After all, each portal has its own design and its own HTML code. For example, at the portal “nur.kz” the content of the news is displayed in the tag. This is to say, the system pulls out all the text content from this tag. Portals www.nur.kz and www.zakon.kz , and portal tengrinews.kz - .

After studying the structure of each news portal, and its architecture, you can proceed to the download process itself. The application algorithm itself looks like this. (fig. 1.):

```
if(tengri!=null&&tengri.equalsIgnoreCase("true")){

    try{

        TrustManager[] trustAllCerts = new TrustManager[]{new X509TrustManager() {
            public X509Certificate[] getAcceptedIssuers(){return new X509Certificate[0];}
            public void checkClientTrusted(X509Certificate[] certs, String authType){}
            public void checkServerTrusted(X509Certificate[] certs, String authType){}
        }};

        try {

            SSLContext sc = SSLContext.getInstance("TLS");
            sc.init(null, trustAllCerts, new SecureRandom());
            HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());

        } catch (Exception e) {

            e.printStackTrace();

        }

        String url = "https://tengrinews.kz/kazakhstan_news/";

        Document document = Jsoup.connect(url).get();

        Element content = document.getElementById("lenta_block");

        Elements links = content.getElementsByTag("a");

        datasetBean.generateTengrienwsDataset(links, operationId, 1L);

    }catch(Exception ex){

        ex.printStackTrace();

    }

}
```

Fig. 1. Part of the implementation of the application algorithm

To implement the download process, we developed a WEB application with a control panel, statistics, and the necessary functionality.

In the “DataSet” download panel, we select all portals and start downloading, which penetrates all news links for today, thereby directly uploading all content from a layout to a specific text array, where later the application will process them and load them into the database.

After that, content was cleared, which would interfere with the machine learning algorithm. In the JAVA implementation, a simple StringTokenizer class is used, which manipulates text. This class has a constructor that specifies characters that can be ignored and used as a delimeter. For example: at the moment the most unnecessary characters are: ^! , ; \ “-

```
StringTokenizer st = new StringTokenizer(text, "^!,.;\"- ");

while(st.hasMoreTokens()){

    String word = st.nextToken();
```

As for the boot process itself, initially the load time from one portal took on average 40-50 seconds. For example, downloading from the site www.tengrinews.kz 171,244 words in one day was 43

seconds. Of course, there is still a factor in the speed of the Internet, in this case we used the usual Internet package IDNET.

The initial load time of about 40 seconds is not bad, but, despite this, we have optimized this process. Since, we are considering, three portals, an average of 2 minutes each, is quite long. In the technical implementation, if you store all the data in the application, in JAVA objects, then it takes some kind of memory and extra time.

Then our application creates and uses about 17,000 objects in one download from the site. But, if you use the entire load at the database level, it is much faster. Many professional developers write SQL procedures at the database level, since indexing, searching and processing text at the SQL level is much faster than at the application level. For example, when authorizing in an application, it is easier to find a login and password at the SQL level from millions of lines at once, than to load all these million lines into an application array, and in a cycle to enter the search for the desired user. Accordingly, we use technology BATCH INSERT [3], which is much faster to insert all the text into the database than if they are loaded one by one. Now, it is possible to download all texts from each portal in one request. The request itself is not complicated and looks like this:

INSERT INTO words (id, operation_id, word, source_id, frequency) VALUES)

VALUES (NULL, 1, ‘law, 1, 1), (NULL, 1, ‘prohibits’, 1, 1), ….

(NULL, 1, ‘development, 1,1);

After the introduction of this approach, the load time has decreased, from each portal it takes approximately 15-20 seconds, 2 times faster.

And also, test downloads were conducted for each day, where the total load time from the three portals took 43 seconds and 41 seconds, respectively.

After successful downloading of all texts for each day, a “DataSet” of unique words is created, using which the machine learning algorithm will be trained, and carry out semantic analysis.

Summary. The purpose of this article is to develop “crawler” for the monitoring system of social moods of society. Here in article we describe the steps of developing a crawler, chosen algorithms and technologies of development and the first experimental results of the work. During the research and experiments, the system will be improved and the results will be published in the following articles.

References:

1. Machine Learning, available at: https://en.wikipedia.org/wiki/Machine_learning

2. Identification of opinion leaders in social networks, available at:

a. https://www.researchgate.net/publication/288427205_Identification_of_opinion_leaders_in_Social_Network

2. WebSockets tutorial, Edition: Tutorials Point (I) Pvt. Ltd. 2016, C. 5-8.

3. Mysql Bible, Steve Suering, Edition: Wiley Publishing, Inc. 909 Third Avenue, New York, NY 10022, C. 150-156