Proc. of Int. Conf. on Current Trends in Eng., Science and Technology, ICCTEST

An Analysis of Data Collection from Social Media and Web

Sumathi M[1] and Dr. Nagaraj G Cholli[2]
[1]Don Bosco Institute of Technology, Dept of IS&E, Bangalore, India
Email: sumahaleh28@gmail.com
[1]R.V College of Engineering, Dept of IS&E, Bangalore, India
Email: nagaraj.cholli@gmail.com

***Abstract*—Contemporary business need is text analytics, Data Collection is the very first step which needs to be accomplished for text analytics. Sources of data are social media, blogs, news articles, emails and search engines, from which the data can be gathered for analysis. Social media sites like Twitter and Facebook holds enormous amount of data which can be turned into business value. Users of social media post their opinions towards any product, person or organization on such social media. Another significant area is search engines, which are also a beneficial source to retrieve information considering any domains. In this paper we discuss the APIs used to gather data from social medias such as Twitter, Facebook and Bing Engines. We are presenting the process of data collection from Twitter, Facebook and Bing Search Engines implementations.**

Index Terms— **Text analytics, Social medias, Java APIs, keyword.**

I. INTRODUCTION

Data collection also termed as information retrieval is the process of gathering textual data and is a preliminary step in text analytics. Text analytics is an important analysis in the fields for prediction, fraud detection and all social media analytics. The data that is accumulated from different sources is in unstructured format. The Twitter API contains four main objects such as tweets, users, entities and places. Users activities are summarized as they can post a tweet, follow other users, create lists and have a timeline, they can be mentioned in other tweets or also can be looked up in bulk. Tweets are the status, opinion updates posted by the users. Some of the actions that we can do in Tweets are embed, reply, like, unlike, re-tweet, delete. Twitter engine provides metadata and additional information about the tweet posted. Places are named locations with geo coordinates. REST API and Streaming API are the two Twitter APIs which can be used to communicate with Twitter programmatically. REST APIs for Twitter provides a programmatic access to Twitter data; supports for both read and write. Streaming APIs provide low latency access to Twitter's global stream. The overhead associated with polling a REST endpoint is avoided using Streaming API. REST API is suited for searching tweets using twitter handles or keywords, to read user profile or to write a tweet. The Streaming API also assist to collect realtime stream of tweets. One of the simple and flexible Facebook Graph API RestFb is written in Java. It is open-source software released under the terms of MIT license.
Bing Search API provides developers to embed the search results from the web using XML or JSON [3]. It

Grenze ID: 02.ICCTEST.2017.1.136

requires AppId which enables API to validate that a request is from a registered Bing application developer. The API interface requires two additional parameters query and sources. Query parameter is the text that is required to execute by the API. Source parameter is the value indicating the source type from where the data is requested.

II. Literature Survey

Rest FB is the API launched by the Facebook and it is an easy way to connect our website with Facebook [1]. RestFB is a simple, flexible Facebook graph API written in Java and is open source software. The features and implementation of Graph API and Old REST APIs are discussed in detail. Graph API is a low level HTTP- based API which can be used to query the data. Facebook can be viewed as a 'social connectivity graph', where the entities, which are users and pages, are considered as nodes and the edges are the connection between the entities. Basic implementation of RestFB with different features and code implementation snippets are discussed.

Twitter is a social media where 140-character short messages called 'tweets' are exchanged [2]. The 'Twitter Streaming API' provides sample of tweets matching some parameters preset by the API user in the query. Tweets were collected for 28 days using the keywords such as Syria, assad, alawite, homas, hama, aleppo, etc., with bounded Geoboxes set to (32.8, 35.9) and (37.3, 42.3) and the user as @SyrianRevo. Tweets were collected using Twitter Streaming API and Twitter Firehose. About 5, 28, 592 tweets were collected from Streaming API and 12,80,344 tweets were collected using Firehose. The two data sets are compared and analyzed. Overall, the results of using Streaming API depends on the coverage and the type of analysis a user or researcher wants to perform.

Ibrahim Toure and AryyaGangopadhyay [4] designed a novel risk projection model to project the terrorism risk levels into the near future. Some of the techniques used in this model are Risk model, Frequency factor, Time factor and Normalization. To predict the terrorist incidents six rules are designed. Here the data is collected from news source and accumulated in database. The data is filtered later and then analysed by different components of the system. The results show that the method can predict within a 1.5 miles radius incident that will occur in the next 24 hours.

III. Implementation

The proposed ideology involves the collection of data from the social media like Twitter, Facebook, and Bing Search engine. Implementation is done on the Eclipse platform using Java as the programming language.

Data Collection from Twitter.

Twitter4j is free and open source software. It is an unofficial Java library for Twitter API. With Twitter4j, the Java application can be integrated with the Twitter service. The first step is to obtain the access tokens for Twitter. Four keys are required – consumer key, consumer secret, access token and access token secret. These keys can be obtained from Twitter Application Manager.

Following is a sample program code snippet which collects existing tweets which contains "Citibank" as the twitter handle from Twitter.

```
ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true);
        cb.setOAuthConsumerKey(consumerKey).setOAuthConsumerSecret(consumerSecret)
        setOAuthAccessToken(accessToken).setOAuthAccessTokenSecret(accessTokenSecret);
Twitter twitter = new TwitterFactory(cb.build()).getInstance();
    Query query = new Query("citibank");
    QueryResult result;
        result = twitter.search(query);
        List<Status> tweets = result.getTweets();
        for (Status tweet : tweets) {
         System.out.println("@" + tweet.getUser().getScreenName() + " - " + tweet.getText());
         }
```

Figure 1. Sample program code snippet which collects existing tweets from Twitter.

ConfigurationBuilder instance is used to setup a connection with the Twitter. Using this instance, the consumer key, consumer secret, access token and access token secret are set. At the time of connection, Twitter verifies all the keys submitted as the valid ones or not. Once the connection is set, Twitter can be queried to obtain desired results. The Query class can be instantiated by setting a keyword, using which the tweets are searched in the Twitter. The result obtained is received in the QueryResult object which can be stored in a List of Status. The List can be iterated over to obtain the individual tweet which is related to the query executed. In Figur.1 above shown example code, executes only one query. A list of queries can be executed by including a loop in the code. The obtained result can then be published in a file for further processing.

Following is a sample program code snippet which exhibits Streaming from Twitter.

```
ConfigurationBuilder cb=new ConfigurationBuilder();
        cb.setDebugEnabled(true);
        cb.setOAuthConsumerKey(ConsumerKey).setOAuthConsumerSecret(ConsumerSecret)
        .setOAuthAccessToken(AccessToken).setOAuthAccessTokenSecret(AccessTokenSecret);
 TwitterStream twitterStream = new TwitterStreamFactory(cb.build()).getInstance();
         StatusListener listener = new StatusListener() {
             public void onStatus(Status status) {
           System.out.println("@" + status.getUser().getScreenName() + " - " + status.getText());
                                                  }
             public void onDeletionNotice(StatusDeletionNotice statusDeletionNotice) {
          System.out.println("Got a status deletion notice id:" + statusDeletionNotice.getStatusId());
                                                                                        }
             public void onTrackLimitationNotice(int numberOfLimitedStatuses) {
                 System.out.println("Got track limitation notice:" + numberOfLimitedStatuses);
                                                                                  }
              public void onScrubGeo(long userId, long upToStatusId) {
    System.out.println("Got scrub_geo event userId:" + userId + " upToStatusId:" + upToStatusId);
                                                                    }
              public void onStallWarning(StallWarning warning) {
                 System.out.println("Got stall warning:" + warning);
                                                                }
    }
    twitterStream.addListener(listener);
          twitterStream.filter("citibank");
```

Figure 2. Following is a sample program code snippet which exhibits Streaming from Twitter.

Connection is setup by using ConfigurationBuilder instance. twitterStream is the class which is used for the implementation of the Streaming API. A Status Listener is available and it has five methods which need to be implemented. Whenever a tweet is tweeted which contains the keyword specified in the filter, the status listener is triggered which invokes the onStatus() method. If a tweet which contains the keyword specified in the filter, is deleted, onDeleteNotice method is invoked. Similarly, onTrackLimitationNotice, onScrubGeo, onStallWarning methods are invoked when the number of tweets collected reaches its limit, the geo location of the tweet is deleted, a stall warning is received, respectively.

Data Collection from Facebook.

Following is a sample program code snippet which depicts the collection of user comments from a Facebook page.

```
FacebookClient facebookClient = new DefaultFacebookClient(accessToken, Version.VERSION_2_3);
Connection<Post> expressFeed = facebookClient.fetchConnection( "citibank/feed", Post.class );
        for (List<Post> expressFeedConnectionPage : expressFeed)
        {
         for (Post post : expressFeedConnectionPage)
```

```
{
Connection<Comment> comments = facebookClient.fetchConnection(id + "/comments", Comment.class);
for(Comment comment: comments.getData()){
System.out.println(comment);
                                                    }
  }
   }
}
```

Figure 3 Sample program code snippet which depicts the collection of user comments from a Facebook page.

An instance of DefaultFacebookClient is created which requires the access token to the Facebook Graph API and the API version. If a valid access token is provided then an object of Facebook Client is created successfully. Requests like comments are returned as a list. Facebook provides a paging mechanism and it can be made use of by calling FacebookClient.fetchConnection() method. The Connection is the result set returned by this method. Connection consists of page full of results which can be iterated over and the comments can be retrieved.

Data Collection from Bing Search Engine

Following is a sample program which shows the data retrieval from Bing Search Engine.

```
URL url = new URL ("https://api.datamarket.azure.com/Bing/SearchWeb/v1/Web?Query=%27
citibank%20");
URLConnection connection = url.openConnection();
connection.setRequestProperty("Authorization", "Basic " + new String(accountKey));
        try (final BufferedReader in = new BufferedReader(new InputStreamReader
(connection.getInputStream()))) {
        String inputLine;
        final StringBuilder response = new StringBuilder();
        while ((inputLine = in.readLine()) != null) {
                        response.append(inputLine);
                        }
        JSONObject jsObj = XML.toJSONObject(response.toString());
        JSONObject feed = jsObj.getJSONObject("feed");
        JSONArray entry = feed.getJSONArray("entry");
                for (int i = 0; i < entry.length(); i++) {
                        JSONObject rec = entry.getJSONObject(i);
String desc =
rec.getJSONObject("content").getJSONObject("m:properties").getJSONObject("d:Description").get
String("content");
                        System.out.println(desc);
```

Figure 4. Sample program which shows the data retrieval from Bing Search Engine

First, an URL connection is established by using the account key for Bing Search API which is obtained from Microsoft Azure Marketplace. The URL consists of the website address of bing marketplace along with the query which needs to be executed. The url can contain any additional parameters such as the site from which data needs to be fetched, language of the data requested and so on. A Buffered Reader stream is used to collect or read data from the connection established. The data retrieved is in the XML format by default. It is converted to JSON object. The JSON object is then examined and the desired field of the object is extracted and processed. The result returned from the Bing consists of one page only. A paging mechanism can be included to collect more data from the Bing Engine.

IV. RESULT

A total of 65,394 data records were collected. The number of data records collected from each of the sources is mentioned in the Table 1. The time period over which the data was collected was four hours

TABLE I. NUMBER OF DATA RECORDS COLLECTED FROM EACH SOURCE

Source	No. of Data Records collected
Twitter	19,048
Facebook	36,589
Bing Search Engine	9,757

V. CONCLUSION AND FUTURE ENHANCEMENT

In this analysis, we discussed some of the APIs used to congregate data from social medias, and presented the implementation of the process of data collection from social medias using APIs. The results of our analysis shows that the comparison of different sources for data records collected for the period of four hours. The APIs used to gather data from different social medias shows that from Facebook source more data collected. For future research, it is planned that to collect additional data for each data record collected.

REFERENCES

[1] JasmeetKaur, Neha Singh, "Facebook Integration with RESTFB API", International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Volume 3, Issue 11, November 2014, ISSN: 2278-1323.

[2] Fred Morstatter, JurgenPfeffer, Huan Liu, Kathleen M. Carley, "Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose", Proceedings of the Seventh International Association for the Advancement of Artificial Intelligence (AAAI) Conference on Weblogs and Social Media, 2013.

[3] Philihp Busby, "Three Different ways to import JSON from the Facebook graph API", Paper SAS379, 2014.

[4] Ibrahim Toure ,AryyaGangopadhyay ,"Real Time Big Data Analytics for PredictingTerrorist Incidents", IEEE, 2016