www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issues 6 June 2016, Page No. 17096-17100

Real Time Analysis using Hadoop

Ramchandra Desai[1], Anusha Pai[2], Louella Menezes Mesquita e Colaco[3]

[1]Goa University, Padre Conceicao College of Engineering,
Verna, Goa, India.
ramchandradesai@gmail.com

[2] Goa University, Padre Conceicao College of Engineering,
Verna, Goa, India.
anusha@pccegoa.org

[3] Goa University, Padre Conceicao College of Engineering,
Verna, Goa, India.
louella@pccegoa.org

Abstract: ***This is an age of BIG DATA. There is huge development in all science and engineering domains which expands big data tremendously. Tweets are considered as raw data and such a huge amount of raw data can be analyzed and represented in some meaningful way according to our requirement and processes. This work provides a way of sentiment analysis using apache hadoop which will process the large amount of data on a hadoop and storm faster in real time. Apache kafka is used here along with apache hadoop and storm as a queuing system. Sentiment analysis of the tweets from the twitter is done to know who the favourite in the tournament is.***

Keywords: Hadoop, Storm, Kafka, Twitter**.**

1. Introduction

The text data on the internet is growing at an enormous pace. Different industries are trying to use this huge raw data for extracting user feedback about their products. Social media is a vital source of information and a good example of such things. It is impossible to manually analyse the large amount of data. That's why there is a need of automatic categorization of such raw data. There are a large number of social media websites that enable users to contribute, alter and grade the content. Users have an opportunity to express their personal opinions about specific topics. The example of these websites include blogs, forums, product reviews sites, and social networks. Twitter tweets are used as the source of raw data. Such sites contain prevalently short comments, like status messages on social networks like twitter or article reviews on Dig.

The twitter data is generated on a huge pace on some news or some events across the globe. Getting the meaningful data from this raw data is a difficult task. There are some open source projects such as Apache Hadoop, kafka, storm, spark, hive which can be used with each other in order to get the meaningful data by analysing the raw data. In this work sentiment analysis on such raw data is done by integrating apache kafka with apache storm and finally storing the scores in apache hadoop (HDFS) for later processing.

2. Related Work

The past research work done in this field have helped us very much. TwitterMonitor [16] explains a system that performs trend detection over a twitter stream in real time in which user interacts with the system by ordering the identified trends using a different criteria and submitting their own description for each of these trend. Real Time Sentiment Analysis of Twitter Data Using Hadoop [1] provides a way of sentiment analysis using hadoop and it uses hadoop cluster for faster processing of huge amount of data. For getting the tweets using the twitter API and getting a fair enough knowledge of HDFS [2] has helped this work fairly as far as getting the understanding of basic knowledge of API's is concerned. The example in this paper explains analysing the tweet as per the geo locations.

Hao Wang and others [5] have described a system for real time analysis of public sentiments towards the candidates of the 2012 U.S. presidential elections as they were expressed on the microblogging site Twitter. Twitter data has been used in prediction of different domains such as social movement's, politics and stock markets [17]. Few analysis found the volume of tweets good enough to predict the 2009 German elections [18]. Some researchers failed to predict the sentiments of the twitter in ranking four candidates in the presidential elections of Singapore in 2011 [19]. There is a blog by Michael Noll [10] which explains the installation of Hadoop as a multi-node cluster. Also a blog [9] by Kunal Gupta was referred in which it explains step by step approach installation of Apache kafka and Apache Storm. These blogs were of greater help is setting up the systems for our work.

Documentation of Apache Storm and Apache Kafka referred from the Apache website has helped us in understanding the concepts of these technologies. A blog by Kenny Ballou [7] has been of great help in development of storm topology and they have explained it step by step approach to implement spout and bolts. [1] Real Time Sentiment Analysis of Twitter Data gives good knowledge of what sentiment analysis is all about and how it can be useful in decision making on broader prospects.

Past studies about analysing the twitter data have been in the fields of politics, economics, events and sports. These studies were basically on the past tweets or done using static samples.

Here it is expected to do real time analysis of the data generated by public which would provide quick indications in changes in opinions.

3. Analysing System

The Large and complex data sets where traditional techniques for data processing were inadequate is termed as big data. Data which gets piled up enormously like in social media sites, enterprise systems on daily basis is unstructured and needs to be processed for meaningful representation.

Characteristics of big data are Volume, Variety, Veracity, Variability and Velocity. Sentiment analysis can help explore how the results of the cricket matches affect public opinions. Actually traditional content analysis takes huge amount of time to complete, the system demonstrated here analyzes sentiment in the Twitter traffic about a particular team, delivering results instantly and continuously.

Data Source

Microblogging site Twitter has been chosen as a source of data. In response to different cricket matches across the globe between the different nations, the volume of tweets goes up significantly and sharply.

Users on Twitter generate over 400 million Tweets every day. Some of these Tweets are available to researchers and practitioners through public APIs at no cost [2]. APIs to access Twitter data can be classified into two types based on the design and access method: REST API's and Streaming API's.

Hadoop

The Apache Hadoop project is an improved open-source software for reliable, accessible, distributed computing.The Apache Hadoop programming takes into account the scattered preparing of substantial information sets crosswise over bunches of systems utilizing basic programming models.

It is intended to scale up from single servers to a great many machines, each offering neighborhood calculation and capacity. As opposed to depend on equipment to convey high-accessibility, the library itself is intended to identify and handle failures at the application layer, so conveying an exceptionally accessible administration on top of a bunch of systems, each of which might be inclined to failures.

Open Authentication (OAuth) is an open standard for verification, received by Twitter to give access to its secured data. Passwords are very defenceless against robbery and OAuth gives a more secure other option to customary confirmation approaches utilizing a three-way handshake. It additionally enhances the certainty of the client in the application as the client's secret key for his Twitter record is never imparted to outsider applications [2].

An application which associates with the Streaming APIs won't have the capacity to build up an association in light of a client demand. Rather, the code for keeping up the Streaming association is ordinarily kept running in a procedure separate from the procedure which handles HTTP tasks.

Hadoop Distributed File System (HDFS) is a dispersed record framework which keeps running on product machines. It is very blame tolerant and is intended for minimal effort machines. HDFS has a high throughput access to application and is appropriate for applications with expansive measure of information. HDFS has a 1 expert server design which has a solitary namenode which manages the filesystem access.

Datanodes handle read and compose demands from the filesystem's customers. They additionally perform square creation, cancellation, and replication upon direction from the Namenode. Replication of information in the filesystem adds to the information honesty and the vigor of the framework.

On the hadoop master node runs the Jobtracker service which monitors the MapReduce task ran by TaskTracker on the slave nodes. User submits the job through JobTracker which then asks the NameNode the location of the data which needs to be processed. Then the Jobtracker locates the TaskTracker on slave nodes and submits the job. Tasktracker accepts the job from the JobTracker thus executing the MapReduce operations.

Let us try to understand real-time systems as far as a Hadoop ecosystem is concerned. This example of the storm topology flow has been taken from blog written by Kenny Ballou. First the storm topology is executed with a classifier provided in the blog i.e. a bag of words model. Implementation of the Naïve bayes classifier was also tried as a part of this work.

Apache Storm and Kafka

Apache Kafka is a messaging queue system which follows the publish-subscribe style of messaging. Zookeeper is used by Apache kafka to save state between all kafka brokers. Each kafka broker maintains a primary and secondary partitions for each topic. Each topic has its partitions circulated over the participating Kafka brokers. The number of times a partition is duplicated for fault tolerance is determined by replication factor. A set of Kafka brokers working together will maintain a set of topics.

Kafka is an appropriated, packaged, imitated submit log mechanism. It gives the usefulness of an informing framework, however with a new outline.

Explained below are the fundamental terms one would run over when managing kafka as shown in fig 1. :

Topics: Kafka keeps up sustains of messages in classifications.
Producers: forms that distribute messages to a Kafka theme.
Consumers: forms that subscribe to themes and process the food of distributed messages.
Broker: Kafka is keep running as a bunch involved one or more servers.

Apache Storm is a real-time computing engine. Also it's made available under the free and open-source Apache license. A topology can be written in any language including any JVM based language, Python, Ruby, Perl, or, with some work, even C. Apache storm runs continuously, consuming data from the configured sources and passes the data down the processing pipeline. Spouts and Bolts combine to make a Topology.

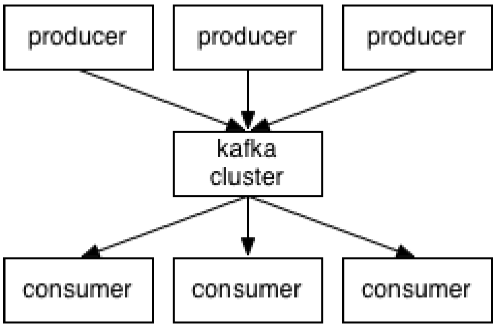

Fig. 1. High Level – Kafka Cluster [5]

Real-time data analytics, online machine learning, ETL are few applications of Apache storm. Storm executes the task with very high speed. It is scalable, fault-tolerant, promises your data will be processed, and is easy to set up and operate.

Apache Storm is a dispersed ongoing huge information handling framework. Tempest is intended to handle inconceivable measure of information in a flaw tolerant and even adaptable technique. It is a gushing information system that has the ability of most elevated ingestion rates. Despite the fact that Storm is stateless, it oversees circulated environment and group state by means of Apache ZooKeeper. It is basic and can execute a wide range of controls on continuous information in parallel [9]. Apache Storm has been a pioneer continuously for information examination.

Fundamentally Hadoop and Storm structures are utilized for dissecting huge information. They two supplement each other and vary in a few angles. Apache Storm does each of the operations with the exception of persistency, while Hadoop is great at everything except for progressive calculation.

Storm can integrate with any queuing and any database system. Some special considerations is required for development of HDFS writing bolt if different writing patterns are considered. There are analysis in which researchers have used Apache flume and hive along with Apache Hadoop.

4. Twitter as a Data Source

A Twitter client's Tweets are otherwise called status messages. A Tweet can be at most 140 characters long. Tweets can be distributed utilizing an extensive variety of portable and desktop customers and using Twitter API. A unique sort of Tweet is the retweet, which is made when one client reposts the Tweet of another client.

Interfacing with the streaming API requires keeping a tireless HTTP association open. As a rule this includes considering the application uniquely in contrast to on and off chance that one connects with the REST API. For an illustration, consider a web application which acknowledges client demands, makes one or more demands to Twitter's API, then organizes and prints the outcome to the client, as a reaction to the client's underlying solicitation.

Following are the key parameters which are used to filter out the tweets which are extracted:

Follow: sequence of userids separated using commas
Track: sequence of keywords separated using commas
Locations: geographical points separated using commas

The real time that is necessary for this work is obtained from the streaming API's delivered by twitter. For the implementation purpose twitter provides streaming API's which allows the developer an access to negligible amount of tweets tweeted at that time based on the particular keyword. The object of which sentiment analysis is performed is presented to the twitter API's which does further mining and provides the tweets linked to only that object. Twitter data is generally unstructured as it uses abbreviations very commonly. Also it allows the use of emoticons which are direct indicators of the author's assessment on the subject. Tweet messages also comprise of a timestamp and the user name. User location if available can also help to gauge the trends in different geographical regions.

The given words in tweet are converted to their root form to avoid the unwanted extra storage of the derived sentiment of the words. The root form dictionary is used to do that and this dictionary is made local as it is widely used is program. This decreases the access time and increases the overall effectiveness of the system.

As the people are aware of the event (match between 2 teams) from the tournament website, the details like team names, the date of match, and the venue and also the most important the player details are visible to outside world. |This information needs to be collected manually. There are many tools such as HootSuite, Tweetchup, Twtrland, Twitter Counter etc. which exist in this domain to analyse and visualize twitter data for different applications.

5. Sentiment Analysis of Twitter Data

Twitter data traffic is maximum during an ongoing match and is minimum in between matches.

Setup

The setup used for this work comprises of 2 ubuntu nodes. Each node is running 64 bit Ubuntu 14.04, given dual core, and 4GB of RAM. Every node is running HDFS (datanode) and Zookeeper. The first node is the namenode, and Nimbus Storm's master daemon. The other node are Storm worker nodes.

Figure 2 shows the system architecture and the different modules. Since this setup is for study purpose, kafka and storm has been installed on the same above machines.

Overview of the storm topology

A simple Kafka producer is written that reads files from the hard-disk which were already downloaded using twitter streaming APIs and sends them to the Kafka cluster. So it uses the offline data since for real time application it needs continuous uninterrupted internet. Incoming messages from the

Kafka brokers are consumed by KafkaSpout. Tweet_id and tweet_text is emitted by the first bolt after parsing JSON data. This implementation only processes English tweets.

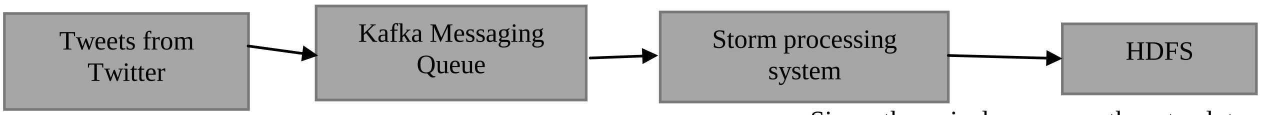

Fig. 2 Data Analysis System Block Diagram

The next bolt performs first round of data cleansing operation by omitting the non-alphabetical characters. In the next round of data cleansing it tries to remove the most common words in order to reduce the noise in future analysis. Such common words are usually known as stop words. The next bolt performs the stemming operation and forwarding the data to the classifiers.

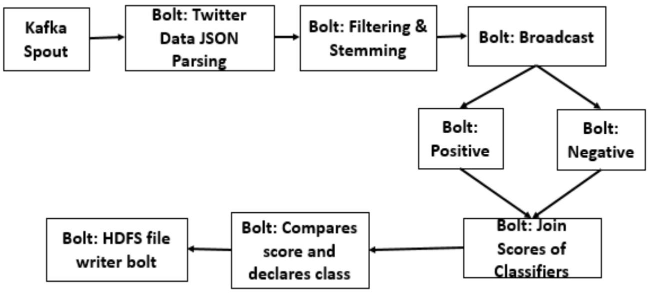
Kafka Spout
Bolt: Twitter Data JSON Parsing
Bolt: Filtering & Stemming
Bolt: Broadcast
Bolt: Positive
Bolt: Negative
Bolt: Join Scores of Classifiers
Bolt: Compares score and declares class
Bolt: HDFS file writer bolt

Fig 3. Storm topology

A separate bolt is developed for each type of classifier i.e. one classifier for the positive category and another for the negative category. Basic classifier has been used in this work.

The scores from the two previous classifiers are compared in the next bolt. Scoring is completed once the bolt compares the scores from the classifiers and specifies the class accordingly. The scoring bolt pushes the results into a HDFS file writer bolt. The HDFS bolt fills a list until it has thousand records in it and then spools to disk. The above figure 3 explains the block diagram of the topology.

Once the scores are stored into the HDFS file they can be processed further or can be aggregated and visualized.

6. Issues with Social Media Content

To make sense from a tweet generated is not that easy. There are lots of challenges which were observed when the content was captured, filtered and processed. Following are few issues which were faced:

- Some compound hashtags, abbreviations or the slang words used are not based on any dictionary or knowledge base.
- Since the allowed length of the tweet is so less that it forces the twitter users to use lots of short forms. For understanding such short forms there is a need of a special dictionary. For example, instead of writing “great” they write “gr8”.
-
- Since there is less space they tend to violate the language rules and thus the traditional information extraction techniques can’t be used.

- Also there are usages of emoticons (e.g. ☺, ;), 😐) and representing their meaning in tweet if the author actually means it or is he/she sarcastically means that.
- Few messages contains only hashtags. Such messages are ignored.

There are few tweets which have similar content which needs to be identified and mark such items as duplicates (which are not considered for analysis). Such tweets may be from a same account. Also the meaning representation of different tweets may be same.

7. Output

The scores from the HDFS file need to have a meaningful representation so that the end user knows like which team is favourite to win the tournament title. Do aggregate the scores for the teams and try to construct a graph similar to the one shown in figure 4. Currently this step is done manually but as a part of enhancement automate this manual process by developing a dashboard for the same.

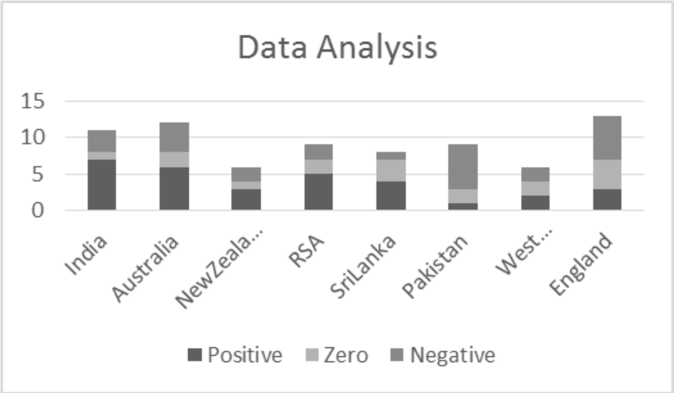

Fig. 4. Output of Data Analysis

When the final scores are into the HDFS, aggregate the scores to visualize the output. The system outputs the number of tweets every minute for each keyword matches. For sentiment, the system outputs the score of positive, negative and Neutral (Zero) tweets. Here positive means the twitter users are supporting their team and feel that they would win. Similarly negative means that they feel the respective team has no chance to win whatsoever. Zero means the author is neutral and has no opinion about the team.

8. Enhancements

Dashboard

Design a HTML based dashboard which will read the scores from the HDFS and depict a visualization of the analysis on the fly which may have drill down options to read the details of the each component.

Usage of Classifier

Implement SVM (Support Vector Machines) and SMO (Sequential Minimal Optimization) classier and do a comparison study to check which one is more accurate in giving results.

Verification of results

It's very difficult as it's not sure if whatever shown as the output is the correct one. Some organisation face this verification task with high security as the data is very confidential and they can't let it out of the organization. Data Security, Scalability, availability and cost needs to be addressed while verifying the results. To test big data, functional and incremental load testing are a must.

9. Data Analysis Applications

Sentiment analysis is used commonly across the world in various aspects. It's up to the data analyst to get the meaningfulness from the raw data in whichever way he or she wants. Basically it's an art of analysis which comes by as you grow in this particular domain.

Our study took a usual approach to examine tweets during several cricket matches and in general. It is required to compare and analogise Indian cricket fans emotional reactions toward both games in which the Indian team played and the games between two non-Indian teams. This analysis, based on sentiment analysis, provided support for the mood theory of sports viewers such that negative emotions increased after one's team loss of a wicket and decreased after one's team scored a boundary. The results also showed that Indian cricket fans, although less concerned about the loss or wickets of non-affiliated teams, enjoyed other games by showing anticipation and joy over the tweets.

10. Conclusion

Sentiment analysis is a very wide-ranging branch for research. A system for analysis of twitter data in real time is explained. Open source technologies like Hadoop, Kafka and storm is used as a part of this system. This system evaluates the sentiments of the people about their favourite cricket team and what happens when a micro event (out, caught, lbw etc.) occur during a match. The methods used and the architecture can be easily used in other domains as it is generic.

Technically the storm topology which determines the classified scores can be enhanced by adding more complex bolts to measure accurate sentiments of the tweets. This work can be extended to other big data sources such as an organisation database and try to analyse the sentiments of the employees within that organisation over certain topics in accordance with the business development team. Also the future work might include analysing the sentiments of the tweets related to sports other than cricket.

References

[1] Real Time Sentiment Analysis of Twitter Data Using Hadoop by Sunil B. Mane, Yashwant Sawant, Saif Kazi, Vaibhav Shinde, College of Engineering, Pune

[2] Twitter Data Analytics by Shamanth Kumar, Fred Morstatter, Huan Liu [Springer]

[3] Apoorv Agarwal, Jasneet Singh Sabarwal, "End to End Sentiment Analysis of Twitter Data"

[4] Apoorv Agarwal, Owen Rambow, Rebecca Passonneau, "Sentiment Analysis of Twitter Data"

[5] A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle by Hao Wang*, Dogan Can**, Abe Kazemzadeh**, François Bar* and Shrikanth Narayanan**

[6] Apache Kafka Documentation: http://kafka.apache.org/documentation.html

[7] Apache Storm Documentation: http://storm.apache.org/releases/current/index.html

[8] Blog by Kenny Ballou: http://zdatainc.com/2014/07/real-time-streaming-apache-storm-apache-kafka/

[9] Storm Tutorial: http://www.tutorialspoint.com/apache_storm/index.htm

[10] Kunal Gupta's blog : http://learnhardwithkunalgupta.blogspot.in/2015/03/apache-storm-installation-in-ubuntu.html

[11] Blog by Maichael Noll: http://www.michael-noll.com/tutorials/

[12] Hadoop, MapReduce and HDFS: A Developers Perspective by Mohd Rehan Ghazia, Durgaprasad Gangodkara

[13] Challenges and Techniques for Testing of Big Data by Naveen Garga , Dr. Sanjay Singlab, Dr. Surender Jangrac

[14] A hadoop based platform for natural language processing of webpages and documents by Paolo Nesi, Gianni Pantaleo, Gianmarco Sanesi

[15] Extracting Semantic Entities and Events from Sports Tweets by Smitashree Choudhury, John G. Breslin

[16] TwitterMonitor: Trend Detection over the Twitter Stream Michael Mathioudakis and Nick Koudas from Computer Science Department, University of Toronto

[17] Twitter mood predicts the stock market by Bollen, J., Mao, H., & Zeng, X.

[18] A sentiment analysis of Singapore Presidential Election 2011 using Twitter data with census correction by Choy, M., Cheong, L. F. M., Ma, N. L., & Koo, P. S.

[19] Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. by Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I.M.