Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

Design of Self-Adjusting algorithm for Twitter Sentiment Analysis

Mr. Amin Nazir Nagiwale
Computer Engineering
Lokamanya Tilak College of Engineering
Navi Mumbai, India

Mr. Manish R. Umale
Computer Engineering
Lokamanya Tilak College of Engineering
Navi Mumbai, India

Mr. Aditya Kumar Sinha
CDAC Pune, India

***Abstract*—Social Media is emerged as an open platform that people use it to share/exchange information. Data captured from social media is generated in different formats, different sources, varies in real time i.e. data generated has big data properties like velocity, volume, variety. For class of machine learning algorithms, large scale data processing frameworks like Hadoop, MapReduce are insufficient as they requires intermediate results across multiple executions. Also they have to dealtwith skewness, diversity of data to adapt changes in real time. For example Social media site, Twitter generates approximately 6000 tweets per second and these are according to trends that changes over time. As success of any machine learning algorithm like Sentiment Analysis is depend upon how well it is trained on training set, domain-specific information and corpus of training set.**

To overcome these problems we have proposed Selfie,Self-adjusting algorithm in [12].In this paper we have discussed preparing domain –specific corpus according to change in trends in real time by using self-adjusting computation. We have implemented Splay tree with trends as tree node and trends near to root of the tree are the most talked/tweeted trends. Each trend node contributes towards domain–specific corpus according to height of tree.

Keywords—Twitter,Self Adjusting Algorithm;Splay Tree; Map Reduce ;Apache Spark

I. INTRODUCTION

MapReduce framework is suitable for data-intensive applications for large scale processing, but there are classes of applications like machine learning algorithms, graph algorithms, sentiment analysis algorithms, etc. have to dealt with skewness, diversity of data to adapt changes in real time. For example, it is difficult to adapt to real time changes in training data/corpus for big data applications like Sentiment Analysis, Email spam detection, and log file analysis.

Also another observed property the Internet demonstrated in the size of its features, we soon discovered a widespread pattern in their measurements: there are many small elements contained within the Web, but few large ones. A few sites consist of millions of pages, but millions of sites only contain a handful of pages. Few sites contain millions of links, but many sites have one or two. Millions of users flock to a few select sites, giving little attention to millions of others Most of the big data computing frameworks assume the data is static i.e. for iterative and interactive applications it must re-execute the entire execution cycle. Due to skewness in the access pattern of user issued queries application data changes incrementally e.g. Consider events like government decisions, launch of new smart phone, release of movie based upon which people start expressing their opinions, sentiments through tweets. Main challenge for Sentiment analysis system is to make system domain-specific. As user's interest is changing from domains like politic, entertainment, marketing, our system has to update its domain specific vocabulary, language corpus according current trends and train their classifiers on newly collected row data i.e. they follow Zipfian distribution[11].

Therefore we have studied and proposed Self Adjusting algorithm, Selfie[12] for class of iterative and interactive applications that will capture the properties of data according to Zipf law [14] by generating Splay tree. Also Self-Adjusting data structure and operations that will help these applications to cache, persist, results, optimize their computation in efficient manner.

For experimental purposes, we have implemented Selfie, self–adjusting algorithm [1] with Splay tree for Twitter Sentiment analysis, which makes system responsible for skewness in access pattern and diversity in trends because skewness in the access pattern of user issued queries application data changes incrementally e.g. Consider events like government decisions, launch of new smart phone, release of movie based upon which people start expressing their opinions, sentiments through tweets. Main challenge for Sentiment analysis system is to make system domain-specific. As user's interest is changing from domains like politic, entertainment, marketing, our system has to update its domain specific vocabulary, language corpus according current trends and train their classifiers on newly collected row data. Therefore proposed algorithm can be helpful for other iterative and interactive applications that faces machine learning challenges like feature generation and selection, over-fitting, explain and improve models to effectively deal with large dynamic data sets.

Proposed system is built on top of Apache Spark [5] andMongoDB[9] harnessing the power of both. Apache Spark will consumes the raw data from various sources and apply user defined MapReduce [7] algorithms on that and processed data is loaded back into MongoDB. MongoDB handles real-time, onlinequeries.

II. ZIPF LAW

Success of any machine learning depends upon how well algorithm is trained i.e. on training set. Preparing accurate training set is really difficult task as we consider size, speed at which data is growing .Capturing the domain-specific data from huge sea of data requires different approach that will respond at real time.

While studying characteristics and patterns of generated data we found that they follow zifian distribution i.e. only 20% of raw data actually contributes.

Consider Twitter sentiment analysis example, twitter generates 5,47,200 tweets per minute, our observations that there is skweness in twitter trends that will change according to events likes Election result declaration, Oscar awards declaration, death of celebrity i.e. what people are talking/tweeting changes according to real time events. So it becomes very difficult to prepare and train machine learning algorithm on domain-specific information. Another observed property is that only few percentage of generated information actually contributes towards specific information.

We proposed solution to this problem by implementing self-adjusting data structure (Splay tree) that respond to the real time changes and capture the required domain specific information according to Zipf law.

According to Zipf law, the frequency of use of the nth-most-frequently-used word in any natural language is approximately inversely proportional to n. Zipf's Law captures the relationship between frequency and rank:

$$f \propto= \frac{1}{r}$$

There is a constant k such that:

$$f * r = k$$

According to Zipf's law probability of most frequently occurring object I over n objects is given by:

$$\frac{1}{\sum_{i=1}^{n} 1/i} = \frac{1}{H_n}$$

and thus the probability of item i occurring is

$$p_i = \left(\frac{1}{i}\right) \cdot \frac{1}{H_n}$$

The proposed a framework for architecture for Sentiment Analysis is show in Fig.1.The framework consists of three major subdivisions Acquisition of data, followed by change initiator and change propagation algorithm.

A. Obtaining real-time/test data

We have used the Streaming Twitter API and the public Streams version of it. All the public data flowing through Twitter currently could be collected using this Public Streams.

B.Change Initiator

This part of system identifies the suitable events such as "Oscar Award Declaration", "Election Result Declarations" ,"Twitter current trend" to initiate change propagation algorithm.

C. Change Propagation

This part of system validates the events raised by the change initiator if access pattern is heavily skewed then change propagation algorithm performs actions like Memorization, self-adjusting data structure management and call the application specific algorithm.

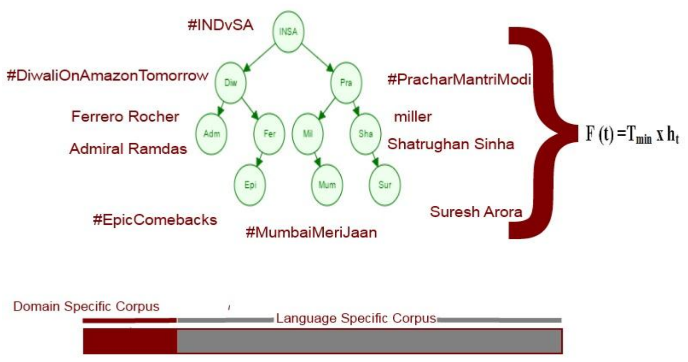

Fig.1. Self-Adjusting framework for Twitter Sentiment Analysis

Proposed system is divided into two major parts:

A.Change Propagation algorithm

B.Application specific algorithm

We have used Splay tree as self-adjusting data structurefor implementation.

Algorithm 1.1: Change Initiator

Input: events, format of the data (csv,log,Real time data)

Output: Pre-processed data for splay tree (Serialized objects or objects of Node class)

```
1: Identify the events and call event handling function
2: Collect data related to the event in formats like
   csv,log,real time data
3: Perform Filtering & smoothing
4: Apply Transformation Operations on pre-processed
   data
```

Algorithm 1.2:Change Propagation

Input: Pre-processed data nodes, Callback events

Output: memorization operation, call for registered events

```
1: Validate the pre-processed data
2: Perform required operations like insertion, deletion,
   split, join
3: Call the registered application specific call back
   event
4: Perform memorization operation on self-adjusting
   algorithm
```

III. Splay Tree:Self adjusting data structure

A splay tree is a self-adjusting binary search tree with the additional property called splaying. Splaying rearranges the tree such that recently accessed element is placed at the root of the tree so that it is quick to access again. Splay tree performs basic operations like insert, delete and search in O(log n) amortized time.

In our proposed system we use Splay tree [1] as a self-adjusting data structure. Its construction process starts by executing Algorithm 1.1*Change Initiator* .Algorithm[12] 1.1 keeps track on events like Twitter Trends and provides this as input to Algorithm 1.2 [12] Change Propagation algorithm.

Formula for Mutant Creation

Suppose that $\{S_i | i \in I\}$ is a set of Splay trees generated at consecutive runs $R_1, R_2 \ldots R_n$then we define:

$$\bigcap_{i \in I} S_i = \{\ x \mid \forall i \in I : x \in S_i\ \} \qquad (1)$$

Where x is splay tree with common nodes from set of splay trees Si.If number of nodes, n(x) = m > threshold value then expand mutant Si where i∈Iotherwise create new mutant S_{i+1} where i∈I.

Algorithm 1.1. [12].provides input to Algorithm 1.2.[12] which will perform user defined function that are based on properties of Splay tree e.g. for Sentiment Analysis it will fetch the tweets based on the height of topic in Splay tree. Topics that are close to the root of the tree will contribute more to training set than topics that are closer to the leaf node.

$$F(t) = T_{min} \times h_t \qquad (2)$$

Where, F =user defined function

t = topic present in Splay tree

T_{min} = threshold value

h_t =height of topic in splay tree

In our Sentiment Analysis example function F contributes to the training set to make train classifier on topics that are currently trending i.e. training set will become more domain specific.

F (#SelfiwithDaughter)>F (#DaagiMantriModike)>

F (#Greece)> …. >F (Zimbabwe)>F (Babulal Guar)

To keeps the track of tweets that are fetched Algorithm 1.2 saves Splay tree, shown in Fig.2.

IV. Results

We have implemented *selfie* algorithm in Twitter Sentiment Analysis to make system domain-specific with respective with respect to change in trends.

We have created training set that is divided into two parts first part contains language-specific (e.g. English) tweets that are randomly collected according to emotions/sentiment present in tweet and second part is dynamically created which contains domain-specific tweets. We have collected 8004 tweets out of which 4002 are positive and 4002 are negative which contributes language specific corpus and remains static.

Experiment is performed on machine with 4GB RAM, processor2.50GHzIntel(R) Core(TM) i5 with 395GB memory and having Ubuntu 14.04 operating system.

Twitter Sentiment analysis , real-time querying and reporting with MongoDB is implemented in Python Programming Language and Algorithm 1.1 [12] and Algorithm 1.2[12] are implemented on cluster computing framework Apache Spark[5] in JAVA programming language. With the help of NLTK metrics module we trained the Naïve Bayes algorithm to calculate precision and recall:

Accuracy	: 0.991
Positiveprecision	: 0.997971602434
Positiverecall	: 0.984
Negative precision	: 0.984220907298
Negative recall	: 0.998

V. Conclusion

We observed collecting domain-specific tweets is difficult because of most of the user uses words that are not part of any standard language corpus. They customize it .In future work we work on these problems also includes feature like polarity weight distribution according to height of the splay tree. We presently implemented Self-Adjusting algorithm for Twitter Sentiment analysis using MongoDB and Apache Spark, experimental results shows that this algorithm is helpful to train the system according to domain specific information available in real-time efficiently.

VI. References

[1] A. Acar. Self-Adjusting Computation. PhD thesis, Department of Computer Science, Carnegie Mellon University.

[2] Programmable Self-Adjusting Computation Ruy Ley-Wild

[3] Streaming Big Data with Self-Adjusting Computation Umut A. Acar, Yan Chen

[4] Parallel Implementation of Big Data Pre-Processing Algorithms for Sentiment Analysis of Social NetworkingData V.Jude Nirmal and D.I. George Amalarethinam.

[5] Spark: Cluster Computing with Working Sets Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica University of California, Berkeley.

[6] Fast Algorithms for Frequent Itemset Mining from Uncertain Data 2014 IEEE International Conference on Data Mining.

[7] Hadoop Map/Reduce tutorial. http://hadoop.apache.org

[8] NoSQL https://en.wikipedia.org/wiki/NoSQL.

[9] MongoDB https://www.mongodb.org/

[10] Twister: Iterative MapReducehttp://iterativemapreduce.org

[11] Theory of Zipf's Law and Beyond Alexander Saichev Yannick Malevergne Didier Sornette

[12] Mr.Amin Nazir Nagiwale, Mr. Manish R. Umale, Mr. Aditya Kumar Sinha,"Design of Self-Adjusting algorithm for data-intensive MapReduce Applications" IEEE conference on International Conference on Energy Systems and Applications (ICESA 2015), IEEE Pune Section