A SURVEY PAPER ON TWITTER DATA ANALYSIS FOR TERRORIST ATTACKS AND VARIOUS ASPECTS

Prof. Anjalidevi Pujari [1], Neha Amolik[2], Varsha Ibitwar[3], Apurva Keskar [4]

[1] *Professor, Department of computer engineering, JSCOE, MH, India*
[2] *UG student, Department of computer engineering, JSCOE, MH, India*
[3] *UG student, Department of computer engineering, JSCOE, MH, India*
[4] *UG student, Department of computer engineering, JSCOE, MH, India*

ABSTRACT

The rapid and tremendous usage of online social platforms such as Facebook, Instagram, Twitter and Others, new APIs are made available for gathering the various social platform information for data collection. Among the all the major social platform the microblogging site Twitter is the most popular one. Twiter APIs allows multiple users to access complete data from Twitter hashtags, accessing to tweets and fetching other useful information such as user tweets and profiles. We are accessing the data from twitter hashtags for analysing the tweets relevant to the terrorism so as to check if any account is associated to terrorist activity in any way. The proposed system makes use for POS tagging for removing the stop words and thereby making the words semantically meaningful. The remaining words are then matched with the synthetic meta-dictionary for terrorism related words. This information gets updated only on a demand basis. The proposed system makes the system robust by not accepting only semantic words but also dynamically written sentences related to terrorism, as the file is make use of contains the complete dataset of terrorism related words.

Keywords:- *Twitter, Micro-blogging, Twitter analytics, Visualization.*

1. INTRODUCTION

Recently, the generalization of micro-blogging (Java et al., 2007; Honey and Herring, 2009) by a wide sector of society- has contributed to transform the way in which people consume information, spread it, and interact with others. The availability of mechanisms to publish short messages, pictures, and videos, together with their use from mobile-phones, allows that information to flow in real-time through a network of users. This change also transforms the way data is processed and integrated in different business models which take into account the information stored in micro-blogging records to determine the next set of decisions to be taken in a business infrastructure (Gudivada et al., 2015; Manyika et al., 2011; Zikopoulos and Eaton, 2011). Twitter is the most popular micro-blogging network.2 It is characterized by the limit in the length of its messages, called tweets (140 characters) and by the asymmetric relations between their users. At the moment, 500 million tweets are published daily, which means a huge source of social data that has caught attention from researchers. From 2008, it is also a source of study by academic researchers (Huberman et al., 2008) and it has been applied in a number of social fields such as elections (Conover et al., 2010; Gayo-Avello, 2011; Barberá and Rivero,2012), social movements (Peña-lópez et al., 2014), predictions (Bollen et al., 2011a; Asur and Huberman, 2010), user's influence (Cha et al., 2010), behavior (Bollen et al., 2011b; Dodds et al., 2011), and message propagation analytics (De Domenico et al., 2013). Currently, it is possible to accede to the complete volume of information on Twitter by means of payment through GNIP.3 Also, researchers may access a partial set of tweets and collect the data by means of specific Twitter APIs. Since its inception, Twitter APIs can be used without any specific restrictions. This favored the creation of services that collected data like tweetbackup.com or Twapper-Keeper.4 However this initial configuration changed a few years ago (September of 2012), with a new rule of use for the APIs that prevented users sharing tweets.5 Currently, the rules of

Twitter's API6 only allow sharing datasets of a limited size. In this context, some tools emerged to process tweets directly from Twitter. Currently, the set of tools that can be used to access tweets includes TwapperKeeper (OBrien, 2011), Twitter-Tap (Kranjc, 2014) and Twitterstream-to-Mongodb (Del Fresno, 2012). However, their specific constraints bring in new concerns and as a result more processing capacity is required. In addition, the specific implementation decisions hamper the type of analysis one may carry out with this type of tools. In a reaction to produce a more open, cost-effective, and flexible approach to Twitter data stream processing, T-Hoarder has been proposed. Additionally, T-Hoarder includes an engine that carries out data analytic processing and displays them in three axes: time, space, and relevance. T-Hoarder is developed to deal with a medium to large number and size of datasets, to gather data for long periods of time (years), to be able to analyse propagations, and to identify the origin of data. Its architecture is also of interest because its underlying ideas can be used in other available engines for stream processing. It also provides solutions to common problems in the way data is processed and/or graphically displayed.

2. LITERATURE SURVEY

2.1 Trendminer: An Architecture for Real Time Analysis of Social Media Text

The emergence of online social networks and the accompanying availability of large amounts of data, pose a number of new natural language processing (NLP) and computational challenges. Data from OSNs is different to data from traditional sources (e.g. newswire). The texts are short, noisy and conversational and the other curious problem is that data occurs in a realtime streams, needing immediate analysis that is grounded in time and context. In this paper we describe a new open-source framework for efficient text processing of streaming OSN data (available at www.trendminer-project.eu). While researchers have made progress in creating or adapting text analysis tools for OSN data, a system to unify these tasks has yet to be built. Our system is focused on a real world scenario where fast processing and accuracy is paramount and we use the MapReduce framework for distributed computing and present running times for our system in order to show that scaling to online scenarios is feasible. We describe the components of the system and evaluate their accuracy. Our system supports easy integration of future modules in order to extend its functionality.

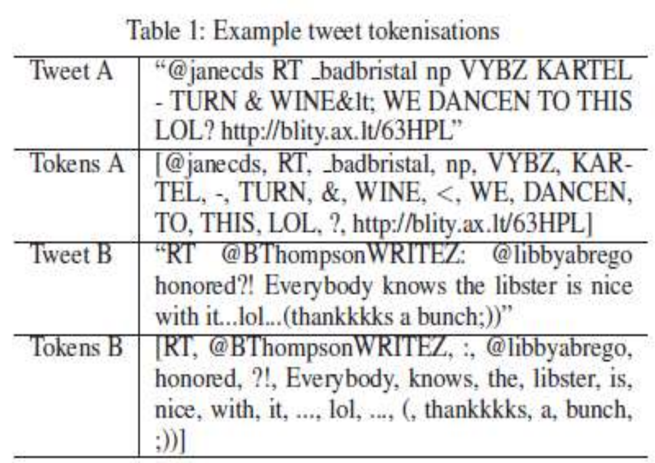

Table 1: Example tweet tokenisations

Tweet A	“@janecds RT _badbristal np VYBZ KARTEL - TURN & WINE&lt; WE DANCEN TO THIS LOL? http://blity.ax.lt/63HPL”
Tokens A	[@janecds, RT, _badbristal, np, VYBZ, KARTEL, -, TURN, &, WINE, <, WE, DANCEN, TO, THIS, LOL, ?, http://blity.ax.lt/63HPL]
Tweet B	“RT @BThompsonWRITEZ: @libbyabrego honored?! Everybody knows the libster is nice with it...lol...(thankkkks a bunch;))”
Tokens B	[RT, @BThompsonWRITEZ, :, @libbyabrego, honored, ?!, Everybody, knows, the, libster, is, nice, with, it, ..., lol, ..., (, thankkkks, a, bunch, ;))]

Figure -1: Sentiment Tokenization in System

2.2 Predicting the Future with Social Media

Recently, social media has become ubiquitous and important for social networking and content sharing. And yet, the content that is generated from these websites remains largely untapped. In this paper, we demonstrate how social media content can be used to predict real-world outcomes. In particular, we use the chatter from Twitter.com to forecast box-office revenues for movies. We view that a simple and easiest model built from the rate at which tweets are created about particular topics can outperform market-based predictors. We further demonstrate how sentiments extracted from Twitter can be utilized to improve the forecasting power of social media.

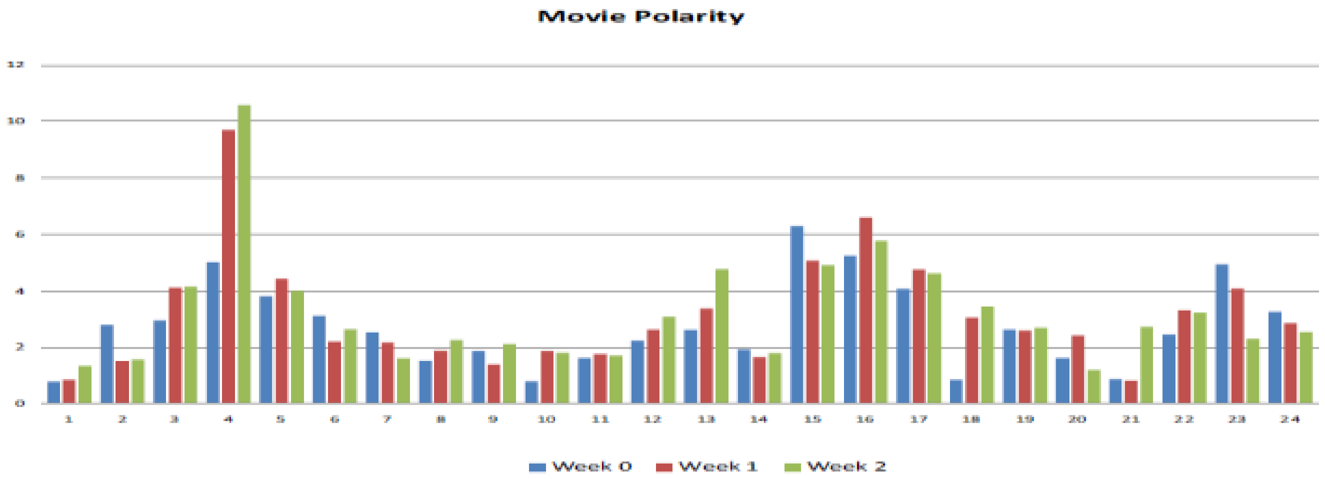

Figure -2: Graphical analysis of Movie review polarity

2.3 Improving the predictability of distributed stream processors

Here, next generation based on real-time applications demand big-data infrastructures to process huge and continuous data volumes under complex computational constraints. This type of application raises new issues on current big-data processing infrastructures. The first problem to be considered is that most of current infrastructures for big-data processing were defined for general purpose applications. A second important limitation is the lack of clear computational models that could be supported by current big-data frameworks and in an effort to reduce and less this gap, this article contributes along several lines. First, it provides a set of improvements to a computational model called distributed stream processing in order to formalize it as a real-time infrastructure and second one, it proposes some extensions to Storm, one of the most popular stream processors. These extensions are developed or design to gain an extra control over the resources used by the application in order to improve its predictability. And in lastly, the article shows few empirical evidences on the performance that can be expected from this type of infrastructure.

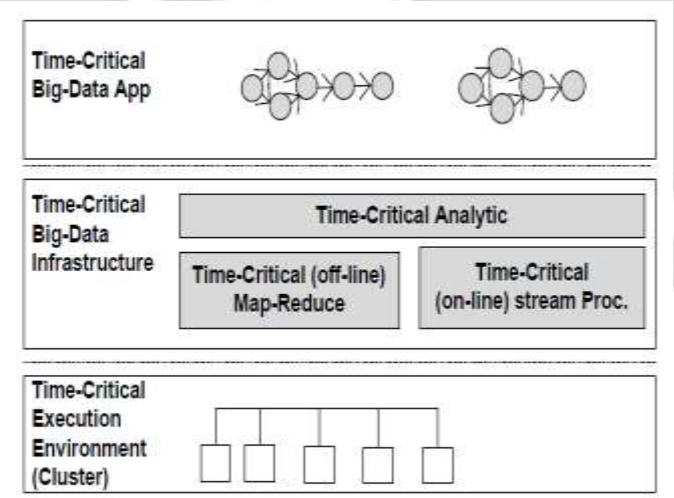
Time-Critical Big-Data App
Time-Critical Big-Data Infrastructure
Time-Critical Analytic
Time-Critical (off-line) Map-Reduce
Time-Critical (on-line) stream Proc.
Time-Critical Execution Environment (Cluster)

Figure -3: Holistic Time-Critical Big-Data System

2.4 Architecting Time-Critical Big-Data Systems

Current effort for designing and developing big-data applications are able to process via big-data analytics- huge amounts of data, using clusters of machines that collaborate to perform parallel computations. However, recent infrastructures were not designed and developed to work with the requirements of time-critical applications; they are more focused on general-purpose applications rather than time-critical ones. Addressing this issue from the perspective of the real-time systems community, this paper considers time-critical big-data and it deals with the definition of a time-critical big-data system from the point of view of requirements, analysing the specific characteristics of some popular big-data applications. This analysis is complemented by the challenges stemmed from the infrastructures that support the applications, proposing an architecture and offering initial performance patterns that connect application costs with infrastructure performance.

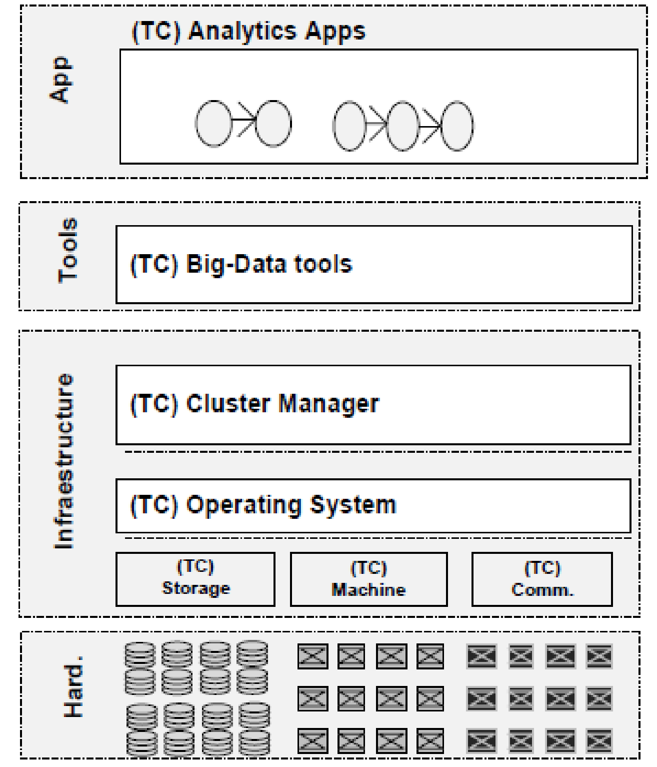

Figure -4: Architecture for Time-Critical Big-Data

3. ARCHITECHTURE

The detail architecture of the system is shown in the below:

Twitter is a popular microblogging service in which users post messages that are very short: less than 140 characters, averaging 11 words per message. It is convenient for research because there are a very large number of messages, many of which are publicly available, and obtaining them is technically simple compared to scraping blogs from the web.

We use 1 billion Twitter messages posted over the years 2008 and 2009, collected by querying the Twitter API,1 as well as archiving the “Gardenhose” real-time stream. This comprises a roughly uniform sample of public messages, in the range of 100,000 to 7 million messages per day. (The primary source of variation is growth of Twitter itself; its message volume increased by a factor of 50 over this two year time period.)

Most Twitter users appear to live in the U.S., but we made no systematic attempt to identify user locations or even message language, though our analysis technique should largely ignore non-English messages. There probably exist many further issues with this text sample; for example, the demographics and communication habits of the Twitter user population probably changed over this time period, which should be adjusted for given our desire to measure attitudes in the general population. There are clear opportunities for better preprocessing and stratified sampling to exploit these data.

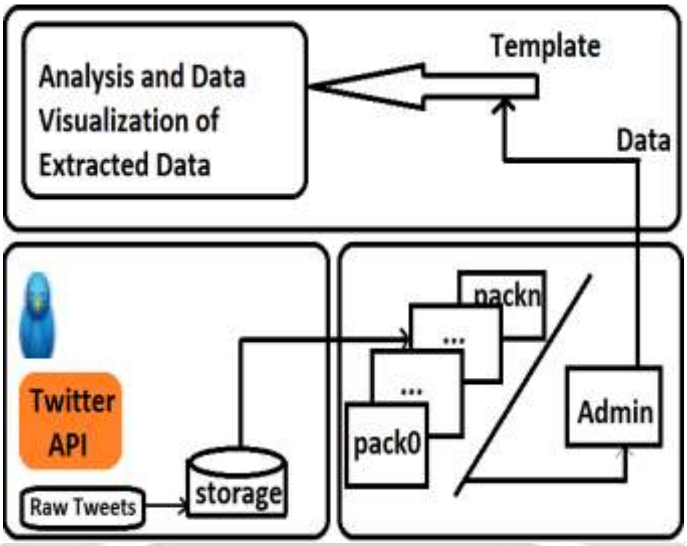

Figure -5: Three levels of T-Hoarder architecture: data collection, processing and visualization.

4. WORKING OF PROPOSED SYSTEM

1. DataCollection:

a. The Data taken as input to the proposed system are the raw tweets extracted from the twitter api into an excel sheet.

b. The excel sheet is given as input to the proposed system and the columns are fetched individual row wise to make the entry of the data in the tabular format.

c. The data being taken as input, being raw data needs to be cleaned and hence the data is given as input to the cleaning module for cleaning of unwanted texts

2. Data Cleaning: The raw data needs to be cleaned for removing the unwanted texts:

a. Extra Punctuation

b. top words (Most commonly used words in a language like the, is, at, which, and on.)

c. Redundant Blank spaces

d. Emoticons

e. URLs

3. Tweet Analysis

a. Once the data is loaded and cleaned, the tweets are then provided as input to the POS tagging sub module for determining the POS of the word available in the tweets.

b. Once the POS tagging of each word is done, the tweets are then analyzed for the meta dictionary words being found for matching the terrorist attack prone words.

c. The tweets which have the meta dictionary words matched or found matching, those tweets are checked for source of tweets and then graphically visualized the % of tweets found to be matched for attack source.

5. MATHEMATICS ASSOCIATED WITH PROPOSED SYSTEM

- Input: Twitter Tweets extracted(T,S,R,C,D,F)
- T: Tweet input
- S: Stop words
- R: Remove Stop words
- C: Classification
- D: Store Database
- F: Final Result
- Output: Stored use tweets to the Database.
- Input: Function Pre-processing (id, request, tweets)
- ID : unique id for each tweet. Request : User request to the server.
- Tweets : user twitter tweets
- Output: Tweet Analysis.

- Input: Function Pre-processingl (id, request, tweets)
- ID : unique id for each tweet. Request : User request to the server.
- Tweets : user twitter tweets
- Output: Tweet Analysis.

6. APPLICATIONS

1. Applications to Review-Related Websites- Movie Reviews, Product Reviews etc.
2. Applications as a Sub-Component Technology- Detecting antagonistic, heated language in mails, spam detection, context sensitive information detection etc.
3. Applications in Business and Government Intelligence- Knowing consumer attitudes and trends.
4. Applications across Different Domains- Knowing public opinions for political leaders or their notions about rules and regulations in place etc.

7. CONCLUSION

The analysis of micro-blogs requires specific tools that help perform global analytics applied to this popular environment. This work contributes with our particular experience in designing and evaluating a tool to perform analytics on micro-blogs. With the aim on Twitter, the article has described a cost-effective framework called T-Hoarder. The main advantage provided by this framework is the possibility of using an integrated approach to store, process, and to display pre-processed data as an output directly on a navigator. The empirical evaluation carried out with several datasets, revealed the performance delivered by the proposed approach which is able to process millions of tweets in seconds.

7. ACKNOWLEDGEMENT

I would like to take this opportunity to express my profound gratitude and deep regard to my guide Prof. Guide Name for his exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. His valuable suggestions were of huge help throughout my project work. His perceptive criticism kept me working to make this project in a much better way. Working under him was an extremely knowledgeable experience for me.

8. REFERANCES

[1]. Preotiuc-Pietro, D., Samangooei, S., Cohn, T., Gibbins, N., Niranjan, M., “Trendminer: An architecture for real time analysis of social media text” 2012, June.

[2]. Asur, S., Huberman, B.A., Computing. Barberá, P., Rivero, G., “Predicting the Future With Social Media,” Desigualdad en la discusion politica en Twitter. Congr. ALICE in 2012.
[3]. Basanta-Val, P., Fernández-García, N., Wellings, A.J., Audsley, N.C., “Improving the predictability of distributed stream processors”, future generation computer systems. SciencieDiret 52, 22–36 in 2015.
[4]. Basanta-Val, P., Audsley, N.C., Wellings, A., Gray, I., Fernandez-Garcia, N., Architecting time-critical big-data systems. In: IEEE Transactions on Big Data, vol. PP, no.99, pp. 1–1. (http://dx.doi.org/10.1109/TBDATA.2016.2622719) in 2016.
[5]. Kranjc, J., 2014. Twitter-Tap. Available: ⟨https://github.com/janezkranjc/twitter-tap⟩.