International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 6, June 2016

Emerging Event Based Tweet Structuring Using Multilevel Inverted List

[1]M.Vinoth, [2] C.Kotteeswaran
[1]Post Graduate Scholar, Computer Science and Engineering, C.Abdul Hakeem College of Engineering and Technology,Vellore,
[2] Research Scholar, Computer Science and Engineering, Sona College of Technology, Salem.
[1]vinothvino017@gmail.com, [2] kottees.svce@gmail.com

Abstract: **Tweet stream are a group of small piece of text which usually represented by the vector space model that constructed on real-life and real-time information. Social network information sharing place the vital role in representing importance of the social network that may dynamically change over time. However to detect and monitor the emerging events from the continuous tweet streams remains a critical factor. Here, I wish to propose a novel indexing scheme called multi-layer inverted list to propagated the emerging events on the social networks (eg: Twitter).Thus, I am in search of facilitated methods to replace the exist in searching mechanism , Cosine similarity method, MIL which combination could give better actuality on detecting and monitoring the emerging events. Extensive experiments have been conducted on a large-scale real-life tweet dataset. The results demonstrate the promising performance of our event indexing and watching ways on each potency and effectiveness.**

***Index Terms*—Event Monitoring, Multilevel Inverted List, Short Text Classification**

I. INTRODUCTION

This online social network is used by millions of people around the world to remain socially connected to their friends, family members. A status update message, called a tweet, is often used as a message to friends and colleagues. A user can follow other users; that user's followers can read her tweets on a regular basis. Twitter is categorized as a micro blogging service. An important characteristic that is common among micro blogging services is their real-time nature. Although blog users typically update their blogs once every several days, Twitter users write tweets several times in a single day. Users can know how other users are doing and often what they are thinking about now, users repeatedly return to the site and check to see what other people are doing.

Many investigators have published their studies of Twitter to date, particularly during the past year to create new applications using Twitter. The real time nature of the updates helps follower's to know about an event. They include social events such as parties, baseball games, and presidential campaigns. They also include disastrous events such as storms, fires, traffic. These events reveal valuable information on breaking news, hot discussions, public opinions, and so on. Moreover, these events are typically evolving over time. Event evolution exhibits event changes across successive time stamps.

The characteristics of tweets involve two main problems to event evolution monitoring. First, the textual content of a tweet is small and noisy, which may disturb the effectiveness of event tracking. Second, incoming tweets arrive in a streaming manner. According to the online statistics, there are around 58 million tweets posted in Twitter per day on average. The event monitoring algorithms for such kind of dynamic social data have to be scalable and incremental without any prior knowledge.

II. RELATED WORKS

Paper [1] presents an investigation of the real-time nature of Twitter that is designed to ascertain whether we can extract valid information (events) from it. It proposes an event notification system that monitors tweets and delivers notification promptly using knowledge from the investigation. In this research, paper take three steps: first, it crawl numerous tweets related to target events; second, It propose probabilistic models to extract events from those tweets and estimate locations of events; finally, we developed an earthquake reporting system that extracts earthquakes from Twitter and sends a message to registered users. Here, we explain our methods using an earthquake as a target event. To classify a tweet as a positive class or a negative class, we use a support vector machine, which is a widely

used machine-learning algorithm. By preparing positive and negative examples as a training set, it can produce a model to classify tweets automatically into positive and negative categories,

But the problem here is processing the instant Hugh incoming text message is very complex task for the above mentioned algorithm which may result in an outdated event information and this paper focus only on a single event so we need a better model for processing the streaming huge data another problem is that we need a proper text processing algorithm to identify the context of the incoming message which are small text message so it should be good at processing short text messages

Another work from [2] which aims to propose and experimentally evaluate an automated system, called Filtered Wall (FW), able to filter unwanted messages from OSN user walls. We exploit Machine Learning (ML) text categorization techniques to automatically assign with each short text message a set of categories based on its content. we base the overall short text classification strategy on Radial Basis Function Networks(RBFN) for their proven capabilities in acting as soft classifiers, in managing noisy data and intrinsically vague classes. In the first level, the RBFN categorizes short messages as Neutral and Non-neutral; in the second stage, Non-neutral messages are classified producing gradual estimates of appropriateness to each of the considered category.

III. PROPOSED WORK

In view of the lack of effective methods for monitoring evolving events from tweet streams, we use four event operations to capture dynamic event evolution patterns, including creation, absorption, split and merge. These four operations are able to track event evolution over time. Further, split and merge operations can also record event relationships in the evolution process. Although similar operations have been mentioned in previous work, our work focuses on how to support these operations efficiently by our proposed indexing structure. To implement fast event search upon the arrival of new tweets, we propose a Multi-layer Inverted List (MIL) as an event indexing structure that can support both efficient event search (for the four event operations) and real-time event update. More specifically, a multi-layer structure is constructed based on the traditional inverted list indexing to support fast search and update for large-scale event databases.

Complete work flow of our proposed scheme in show in the fig.1 The inverted index data structure is a central component of a typical search engine indexing algorithm. A goal of a search engine implementation is to optimize the speed of the query: find the documents where word X occurs. Once a forward index is developed, which stores lists of words per document, it is next inverted to develop an inverted index. Querying the forward index would require sequential iteration through each document and to each word to verify a matching document. The time, memory, and processing resources to perform such a query are not always technically realistic. Instead of listing the words per document in the forward index, the inverted index data structure is developed which lists the documents per word. With the inverted index created, the query can now be resolved by jumping to the word id (via random access) in the inverted index.

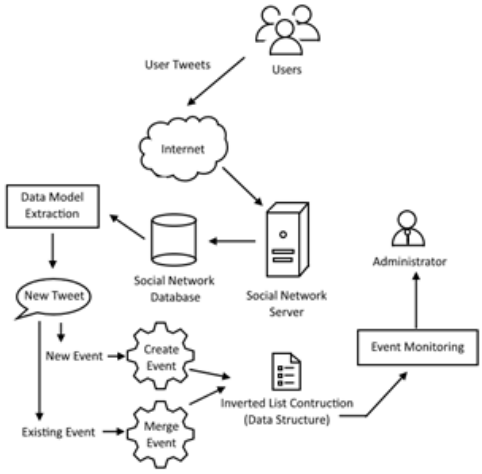

Fig.1 Architecture Diagram

A. Data Model Creation

Event prediction process need meta information like date of the post, time of the post, location of the post etc. from the processing tweet for analyzing the events so we need to construct a data model which these event meta information extracted real time tweets. In data model creation phase our system process both small textual post and the meta information of a tweet for extracting the data model because the users may post a particular event which may have different timestamp so he / she may mention the difference in time within the post so the timestamp and location information mentioned inside the post has more priority than the timestamp and location information in the post meta information. *Fig.2* explains the process flow of the data model extraction process

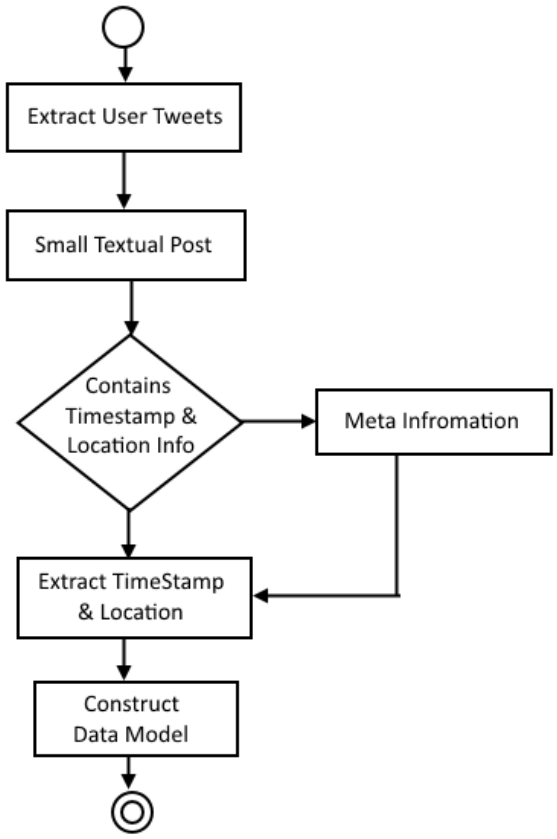

Fig.2 Data Model Extraction

B. Cleaning Data Model

In human nature of representing a same event or information in a different manner adds more complexity to our data modeling phase because processing document or text with different works and similar context is very difficult. We need to clean the extracted data model so that we can classify the similar event correctly. We use word semantic to understand the relationship between words for eg. hurricane and storm are different word representing same event. Semantic analysis removes this problem from our data model.

Table 1: Data model before semantic process

Event Id	Event	Date	Location
E1	Storm	2000/02/02	Chennai
E2	Earth Quake	2009/11/03	Chennai
E3	Hurricane	2000/02/02	Chennai

Table 2: Data model after semantic process

Event Id	Event	Date	Location
E1	Storm	2000/02/02	Chennai
E2	Earth Quake	2009/11/03	Chennai
E1	Hurricane	2000/02/02	Chennai

C. Inverted List Construction

Normally social networks receive thousands of tweets every day, so the extracted data model will also large in size. Process this large data model is very costly so we create an inverted list which reduce the cost of processing the huge data model. The purpose of an inverted index is to allow fast full text searches, at a cost of increased processing when a document is added to the database. The inverted file may be the database file itself, rather than its index. It is the most popular data structure used in document retrieval systems used on a large scale for example in search engines.

D. Event Create & Absorb

Even after construction of inverted index list we will be receiving new tweet for even new time instances, so we need to update the index structure frequently. For update process we need an optimal way to update the index structure will less cost. If we don't find similar event within the index will create a new node within the index structure if we find an existing event within the index we will insert that post into that node. thus index node is updated based on new incoming posts.

E. Event Splitting & Merging

While processing the text there may be some deviation in the predicted event (i.e) we may have sub event predicted within an event which bring in need of splitting and in some cases apart from the semantic analysis we may has two different event representing a single event so we also need to merge this two events into one single event this module is used by our system to split and merge the event for more accuracy of event monitoring process.

F. Event Monitoring

Our event monitoring phase include an event detection module which produce an alarm when the event reaches an acceptable threshold set by the data analysist. The leaf nodes in the index structure hold both the event label and the weight of the detected event, when the weight of the detected event exceeds the set threshold a notification is given. To receive the notification the users need to subscribe for the particular events

IV. CONCLUSION

In this paper, we have proposed a new event monitoring scheme with semantic analysis and inverted indexing structure to efficiently and effectively index evolving events from tweet streams. Temporal information is a significantly extracted using our data model extraction process which is a main effecting factor of event monitoring in social data like tweets. Semantic analysis process reduces the noise from the extracted data model which avoids the need of further analysis of constructed index. Four operations are designed to capture the dynamics of events over time.

REFERENCE

[1] Takeshi Sakaki and Makoto Okazaki, "Tweet Analysis for Real-Time Event Detection and Earthquake Reporting System Development", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, PP. 4, April -2013.

[2] Marco Vanetti, Elisabetta Binaghi and Elena Ferrari,"A System to Filter Unwanted Messages from OSN User Walls", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, PP. 2, February -2013.

[3] Hongyun Cai, Zi Huang and Divesh Srivastava,"Indexing Evolving Events from Tweet Streams", IEEE Transactions On Knowledge And Data Engineering, Vol. , PP. 4, April -2015.

[4] Y. Jie, L. Andrew, C. Mark, R. Bella, and P. Robert, "Using social media to enhance emergency situation awareness," IEEE Intelligent Systems, vol. 27, no. 6, pp. 52–59, 2012.

[5] S. Unankard, X. Li, and M. Sharaf, "Emerging event detection in social networks with location sensitivity," World Wide Web, pp. 1–25, 2014.

[6] A. C. Awekar and N. F. Samatova, "Fast matching for all pairs similarity search." in Web Intelligence, pp. 295–300,2009.

[7] A. Uszok, J.M. Bradshaw, M. Johnson, R. Jeffers, A. Tate, J. Dalton,and S. Aitken, "Kaos Policy Management for Semantic Web Services," IEEE Intelligent Systems, vol. 19, no. 4, pp. 32-41, July/Aug. 2004.

[8] M. Sarah, C. Abdur, H. Gregor, L. Ben, and M. Roger, "Twitter and the Micro-Messaging Revolution," technical report, O'Reilly Radar, 2008

[9] K. Borau, C. Ullrich, J. Feng, and R. Shen, "Microblogging for Language Learning: Using Twitter to Train Communicative and Cultural Competence,"Proc. Eighth Int'l Conf. Advances in Web Based Learning (ICWL '09),pp. 78-87, 2009.