International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P), Volume-6, Issue-3, November 2016

# Clustering and Summarization with Timelines on tweet streams

**Dr B Rama, Kande Archana, G. Prabhakar Reddy, K. Mounika**

***Abstract*— In recent days usage of social networking sites like twitter, face book and Instagram has become so popular that they allows us to become informed on various aspects of companies, news, people, their views and emotions. Tweets on twitter arrive at a very high rate of 6000 tweets per second and are created in unorganized and informal way with lots of noise and redundancy. The searching process for concerned 140 characters of short message (tweet) out of the huge data collection becomes a hectic task to both users and data analysts. The best solution which we can frame for this problem is to apply continuous summarization technique on it with TCV Rank summarization algorithm after initially clustering the tweet streams using k-means clustering. Summarization along with automatic timeline generation with the topic evolution detection method algorithm minimizes the efforts. This technique when implemented will make the work of data analysts and also end user easier specially in searching important tweets. This paper mainly focuses on efficiency, flexibility and topic evolution factors.**

***Index Terms*— Clustering, Summarization, Tweet stream, Timeline generation.**

## I. INTRODUCTION

In this modern world micro blogging services have gained lots of importance in people's life. This growth in its popularity has in return resulted in the increase in the short text messages flood on twitter, face book, tumbler, etc. As per 2016 stats it is observed that over 313 million people all around the world use twitter as an invaluable source of news, blogs, opinions and more. When we try to search a concerned topic on twitter, it retrieves thousands of contents which includes that particular content, but most of the content retrieved (tweets) is noisy, meaningless and irrelevant due to the social nature of blogging. Other than that the time taken to retrieve this required content is extent spanning to many hours. To make this search much more worse, the newly arriving tweets which satisfy the search criteria keep flooding at a very high rate. The only best solution for this problem is summarization. Summarization represents to restate the main idea in few words as possible. Summarization of a topic is done after clustering process where the messages are scattered in to the respected groups related to it. Summarization reduces the redundancy by including the main and sub topics in it. Summarization approach was firstly adopted for documents, where the data was static, and had small data collection. The same approach is used for dynamic data generation in twitter but it could not satisfy its requirements due to the fast arriving tweets and its large data collection. Thus modifying the summarization nature of the traditional approach in terms of its functionality is a must since tweets are strongly related with their posted time. Consider a scenario where the user is interested in a particular topic say, cricket. When the search is made on twitter with that keyword Cricket, all the data containing the word is retrieved with respect to the timelines specified by the user. The growing demand for summarizing large contents in social services fuels the development in visual techniques. Time line is one of these techniques which can make analysis process easier and faster. Adding this feature of timelines to the summarization helps in highlighting the points which include the topic and sub topic related to the search and ignores the period other than required. This helps the user to Zoom in(Aug 3, 7Am to 10 Am)and Zoom out (Aug 3 to Aug 9)in to the topic. **In this paper we propose a new summarization technique called continuous summarization addressing the features of efficiency on large data sets, flexibility by providing summaries on random time durations, topic evolution factors.**

## II. LITERATURE REVIEW

Zhenhua Wang introduced a summarization framework called Sumblr This is the continuous summarization by stream clustering[1]. CluStream [2] method of clustering used an online clustering component and an offline clustering component. PTF data structure was used to recall the historical micro clusters for different time frame. Clustering process which is the first step before summarization was carried by different authors differently. BIRCH adopted an in-memory data structure called CF- tree for clustering [3]. Bradley used a technique of storing the important portions of the data and discarded the rest [4]. It is a scalable clustering mechanism. Partition based approach was adopted to resolve the problems related to filtering, text crawling, topic detection, but still failed to address clusters formed over different time durations. Authors in [5] used clusStream and extended its features. Time based clusters were generated with both online and offline phases. Our tweet steam clustering method has only online phase using a data structure called TCV [5]. Micro blog / Document summarization can be extractive or abstractive in nature. They basically concentrate on extractive approach where selective statements are selected. Ranking approach is adopted in most of the existing algorithm for document summarization. Phase reinforcement algorithm is used to summarize tweet posts using a single tweet [6]. Hybrid TF-IDF algorithm and cluster based algorithm was used to generate multiple post summaries [7]. All the above methods use small and static data and are not effective in efficiency and evolution factors. Identification of participants of events and generate summary based on sub events is proposed in [7]. They store all the distilled statics of tweets in a data structure called TCV instead of segments as used in traditional methods for

historical summarization. Usage of TCV snapshots helps in online summarization as well. Efforts were made in the timeline generation area in 2008 by Diakopoulos and Shamma in [9]. Timeline based backchannel was proposed in [10]..

## III. Existing System

When there are millions of Tweets arriving every day, searching for a hot topic in Twitter may yield huge collection of tweets, spanning weeks. Even when Filtering technique is used, going through so many tweets for important contents would be impossible and the enormous amount of noise and redundancy is much more irritating. Along with that, new tweets satisfying the filtering criteria may arrive continuously, at very high rate making things much worse. Moreover, tweets are strongly correlated with their posted time and new tweets tend to arrive at a very fast rate.

Unfortunately, existing summarization methods cannot satisfy today's requirements because they mainly focus on static and small-sized data sets, and they are neither scalable nor efficient for large data sets. It is not acceptable to continuously summarize for every possible time duration. Their summary results are insensitive to time. Topic evolution is also a tough task for them.

## IV. Proposed Methodology

In this paper we propose a component based structure with summarization and Time line generation as main modules. Summarization which is in return a two step process with clustering at its first stage. We adopt an online clustering algorithm for effective clustering with only one pass over the data. This clustering process used two different data structures to store the tweets namely Tweet cluster vector(TCV) and Pyramidal Time Frame(PTF).Now come the summarization process which supports two modes, Online and historical. Online summarization uses TCV- rank summarization algorithm to compute centrality scores for tweets stores in TCVs. Historical summarization where the snapshots are retrieved from PTF base on time duration and compared. Next is the Timeline generation process where topic evolution detection algorithm is used to generate real-time / range timelines based on 2 kinds of variations, volume based and content based.

## V. Detailed Architecture

Summarization which is the process of describing the main idea of the whole content in few words can generate online summaries and historical summaries as well. Online summaries are framed on what is currently discussed among people and thus the input for this summarization process is directly taken from the clusters stored in the memory. Historical Summaries, helps people to understand the hot topics happening in that particular period. Thus it is a very complicated task. Let's assume that the length of a user defined time duration is s, and the ending time stamp is t. From PTF we can retrieve two snapshots whose timestamps are either equal to or right before t and t-s. TCV Rank Summarization algorithm is used to select representative tweets to form summaries. Here comes the concept of TCV which is the data structure used to store the tweets.TCV is used as a data structure in clustering. Clustering is the initial step to be performed before summarization. It uses a novel compressed structure called Tweet cluster vector (TCV) which are stored in memory dynamically during stream processing. Its definition is an extension of cluster feature vector which keeps information of tweet clusters.TCV stores the original tweets and also temporal information. Our TVC structure can also be updated in incremental manner when new tweets arrive. PTF is the second data structure which we use to store maintained TCVs at particular moment called snapshots. PTF(Pyramidal Time Frame) stores snapshots at different level of granularity depending on the regency. Fig.1 gives a complete view about the architecture.

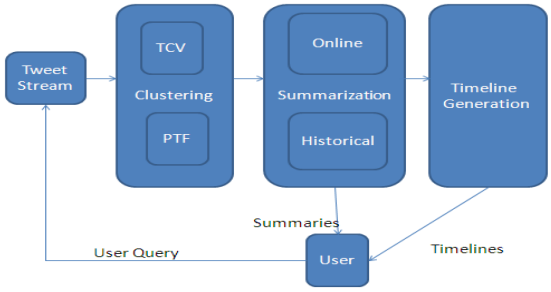


Fig.1 System Architecture

### *A. Incremental Tweet Stream Clustering Algorithm*

Online statistical data is maintained by the tweet stream clustering module. K-means clustering algorithm is applied on the tweets to create clusters and TCVs are initialized accordingly. The tweet stream clustering process keeps updating the TCV whenever a new tweet arrives. Consider that a new tweet (T) arrives at a time(t) and there are (n) cluster active by then. We firstly need to decide whether we can add this new tweet in to the existing clusters or we need to create a new cluster (nc) respectively for that tweet. For this we initially figure out which existing cluster(c )'s centroid is closer to that newly arrived tweet. We find the cluster with the Maximum Similarity (MS). Even though the centroid of a particular cluster is closer to a tweet that tweet may not be completely related to that cluster topic, thus we find the Minimum Boundary Similarity (MBS). MBS is the average closeness between the centroid and the tweets included in cluster c. If MS is less than MBS then new cluster(nc) is created else tweet is added in to the existing cluster(c)of cluster set(C).When the current time stamp(ta) is divisible by (xi) for a integer(i), we store the snapshot of current TCVs in the disk and index it by PTF.

Input: A cluster set C
STEP 1: while ! Tweet_stream.last()
do T = stream.next()
STEP 2: Choose c in C whose centroid is the closest to T
STEP 3: if MS < MBS then,
create a new cluster nc = {T}
STEP 4: C.add(nc)
Else
c.add(T)
STEP 5: if ta % xi = = 0 then PTF.add( C )

Output: Newly created clusters and updation in existing clusters.

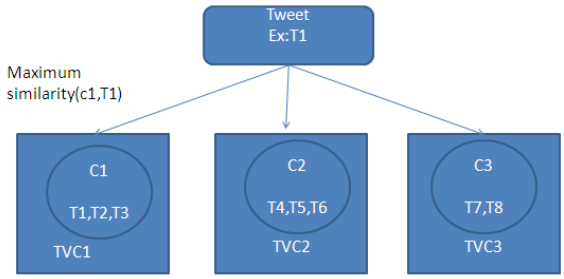


Fig.2 TCV Algorithm

Maximum Similarity(C1,T1)<MBS = update T to a new cluster.
Maximum Similarity(C1,T1)>=MBS = update T to a closest cluster.
Once the clustering is done and respective clusters vectors are created the summarization process starts. Cluster vectors which are formed as per the topic are analyzed and summaries are generated accordingly. Searching process can be made easier through summarization. Fig.2 Describes more about TCV algorithm.

### *B. TCV Rank Summarization Algorithm*

Consider an input cluster set C, TCV set as D(C). Tweet set TS consists of all the tweets in FS in D(C).The main problem here is to extract k tweets from T.

Here F = {T1,T2,….,Tt} which is a collection o9f non empty subsets. Here Ti represents a sub topic. |Ti| represents the number of its related tweets.

Input: a cluster set C
STEP 1: S = Ø . T ={ all the tweets in ft sets of D(C) }
STEP 2: Build a similarity graph on T;
STEP 3: Compute LexRank scores LR;
STEP 4: Tcluster ={tweets with the highest LR in each clusterg}
STEP 5: while |S| < L
do
foreach tweet ti in Tcluster - S
do
calculate vi
STEP 6: select tmax with the highest vi;
add(thig)
while |S| < L
do
foreach tweet ti in T - S
do
calculate vi
STEP 7: select thig with the highest vi
STEP 8: S.add(thig)
Output: a summary set S

Addition of timeline feature makes this process much more flexible, faster and efficient, as the filtering is done based on time duration as specified by the user.

### *C. Time Generation Algorithm*

Topic evolution detection method produces real- time and range timelines in similar way. During the stream processing the subtopic changes can be monitored and discovered. Basically we divide the tweets by day/time as the tweet proceeds and this sequence is taken as input for the algorithm. Whenever a new variation is found new timelines are appended.

Input: a binned tweet stream
STEP 1: set n=Ø
STEP 2: while !stream.last()
do
Bin bx = stream.next()
STEP 3: if hasLargeVariation()
then
n=add(x);
STEP 4: return n;
Output: a timeline node set n

A new node on the timeline can be created when sub topic change occurs and a particular variation is found. This variation can be observed by considering 2 factors.
Summary based variation: A tweet stream is summarized when arrived and form online cluster statistics in TCV's. This helps in generating the timelines.
Volume based Variation: The use of rapid increase(spike) resembles the increase in the volume of the tweets over time.

## VI. EXPERIMENTAL RESULTS

Our experiment shows the implementation of all the 3 main topic of the paper- clustering, summarization and time line generation by constructing a mini system which resembles a tweeter portal with sample data. This implementation showcase the whole framework in terms of both data analyst(admin) and end user. Initially the user creates his/her profile in the portal and the admin accepts/cancels the request. Now the user logs in to the system and starts tweeting or views and comments on already existing tweets which in-turn are again tweets. The below class diagram gives a complete picture about the functionalities carried out by each role.

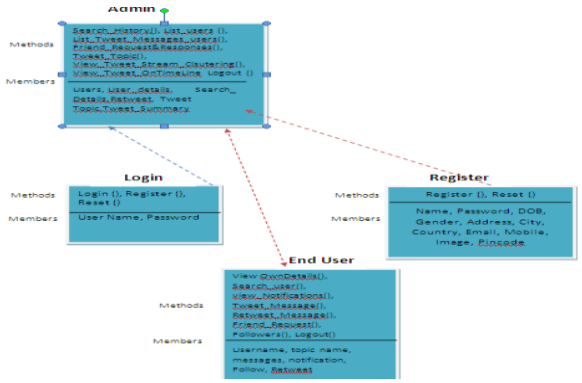


Fig.3 Experimental Overview

| Topic Name | Tweeter Name | Tweet | Date | Rank |
|---|---|---|---|---|
| elnfield gt500 | sagar | It is very Power Full Bike | 19/10/2015 17:43:06 | null |
| elnfield gt500 | sagar | red color looks good | 20/10/2015 12:26:56 | null |
| infield classic350 | sagar | very good to ride | 30/10/2015 15:33:37 | null |
| sachin | varun | he is a good player | 20/10/2015 11:49:17 | 1 |

Fig.4 Experimental Result

The concept of clustering is implemented with few input tweets which are clustered respectively in to their clusters based on the closeness of each tweet with the cluster topic. Summarization is done with the rank generation on each tweet based on the response of the end users and finally the tweets along with their time slots is also captured at which they are created through which the searching process gets easier. The below graph gives a detailed pictorial representation of the variations in the number of tweets arriving at different time bins in a month.

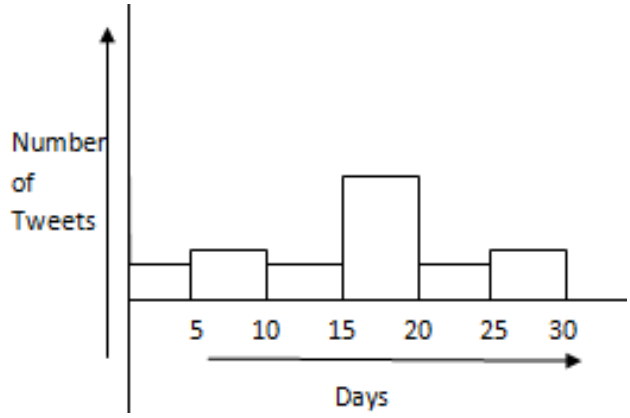


Fig.5 Graphical view of arriving tweets rate

## II. Conclusion

The proposed system supports continuous tweet stream summarization after online clustering of the tweet streams by compressing the tweets stored in TCV data structure and snapshots stored in PTFs. Then TCV_ Rank summarization algorithm generated online and historical summaries by considering the specific time duration. Now, the evolutionary topic detection method algorithm is applied on the summaries which automatically generate the time lines for the tweet streams. This method addresses efficiency, flexibility and scalability factors. This work can further be extended an implemented for distributed systems. The same can be implemented for multi-topic version as well.

## References

[1] Zhenhua Wang, Lidan Shou, Ke Chen, "On Summarization and Timeline Generation for Evolutionary Tweet Streams", in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO.5,MAY 2015.

[2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in Proc. 29th Int. Conf. Very Large Data Bases, 2003, pp. 81–92.

[3] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1996, pp. 103–114.

[4] P. S. Bradley, U. M. Fayyad, and C. Reina, "Scaling clustering algorithms to large databases," in Proc. Knowl. Discovery Data Mining, 1998, pp. 9–15.

[5] C. C. Aggarwal and P. S. Yu, "On clustering massive text and categorical data streams," Knowl. Inf. Syst., vol. 24, no. 2, pp. 171–196, 2010.

[6] B. Sharifi, M.-A. Hutton, and J. Kalita, "Summarizing microblogs automatically," in Proc. Human Lang. Technol. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2010, pp. 685–688.

[7] D. Inouye and J. K. Kalita, "Comparing twitter summarization algorithms for multiple post summaries," in Proc. IEEE 3rd Int. Conf. Social Comput., 2011, pp. 298–306.

[8] C. Shen, F. Liu, F. Weng, and T. Li, "A participant-based approach for event summarization using twitter streams," in Proc. Human Lang. Technol. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2013, pp. 1152–1162.

[9] N. A. Diakopoulos and D. A. Shamma, "Characterizing debate performance via aggregated twitter sentiment," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2010, pp. 1195–1198.

[10] M. Dork, D. Gruen, C. Williamson, and S. Carpendale, "A visual backchannel for large-scale events," IEEE Trans. Vis. Comput. Graph., vol. 16, no. 6, pp. 1129–1138, Nov. 2010