ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

Summarization and Sentiment Analysis of Social Content

Bhagyashri B. Kamankar [1], Mr. Manesh P. Patil[2]

P.G. Student, Department of Computer Engineering, SSVPS BSD COE, Dhule, India[1]

Assistant Professor, Department of Computer Engineering, SSVPS BSD COE, Dhule, India [2]

ABSTRACT: Tweets are being created short text messages and shared for both users and data analysts. Twitter which receive over 400 million tweets per day has emerged as an invaluable source of news, blogs, opinions and more. These short messages, are very informative, but at the same time they are very overwhelming. Our proposed work consists three components tweet stream clustering to cluster tweets using agglomerative cluster algorithm. And second tweet cluster vector technique to generate rank summarization i.e. tweet summarization and third to detect and monitors the summary-based variation to produce timeline automatically from tweet stream. So proposed system achieves efficiency i.e. summarization of large data set, flexibility i.e. system provides summary of arbitrary time duration and topic evolution i.e. system routinely detects sub - topic changes and the moments that they happen. In addition we design sentiment analysis i.e. polarity and Topic popularity of tweet stream. Our experiments gives efficiency and effectiveness on large scale data set.

KEYWORDS: Tweet stream, popularity of topic, summary, sentiment analysis, etc.

I. INTRODUCTION

Twitter gained popularity among all microblogging services. Twitter has millions of users and therefore it has high popularity amongst people. These sites which receives millions of short text messages per day. These short-text messages such as tweets, facebook comments, etc. are a very good medium for message transferring or message giving. The use of social media has brought the world very closer. People can get opinions and suggestion on different topics from these sources. There are many reviews, debts and discussion happens on social groups. Most of important topics are being shared on social accounts. To make things worse, new tweets satisfying the filtering criteria may arrive continuously, at an unpredictable rate.

One possible solution to information overload problem is summarization. Summarization [1] represents restating of the main ideas of the text in as few words as possible intuitively, a good summary should cover the main topics or subtopics and have diversity among the sentences to reduce redundancy. Summarization is widely used in comfortable arrangement, especially when users surf the internet with their mobile devices which have much lesser screens than PCs. Traditional document summarization approaches, however, are not as effective in the situation of tweets given both the big size of tweets as well as the fast and continuous nature of their arrival [1].

Summarization represents a set of documents by means a summary consist of short description of the whole data. Summary covers main headings and sub headings. Summarization is the process of reducing a text document with a computer program for creating a summary that contains the only important points of the original document. The problem of information overload is increases, and because of the quantity of data is increasing, there is a necessity automatic summarization. Machine learning and data mining in which automatic data summarization is a very important area. These summarization technologies are widely used today, in a large number of micro blogging industries. The main idea behind summarization is to search a representative and common subset of the data, which represent unique information of the entire set. Document summarization is classified as extractive and abstractive

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

summarization [3]. Abstraction involves paraphrasing sections of the source document. Extractive performs the automatic system extracts objects from entire collection, without modifying the objects themselves.

In this system, we propose a novel tweet streams. We first propose a tweet stream clustering algorithm to cluster tweets and maintain distilled statistics called Tweet Cluster Vectors. In existing base paper, at the start of the stream, k-means clustering algorithm used to create the initial clusters. With global cluster, it didn't work well. In our proposed work, we use agglomerative clustering [4]. Then we develop a TCV summarization technique for generating summary of arbitrary time durations. Finally, we describe a topic evolution detection method, which consumes current summaries to produce timelines automatically from tweet streams.

The system can also be provided with few extra features such as obtaining sentiments of the tweets and also calculating the popularity of that particular topic. The system uses text mining algorithms to obtain the desired results. The major focus of the system is to generating summary and obtaining sentiment of that tweets according to user preferences.

II. LITERATURE SURVEY

Viewing large amount of tweets which pop up in short frame of period is not an easy task, since a large number of tweets are meaningless, irrelevant and noisy in nature, due to the social nature of tweeting. Further, new tweets tend to arrive at a very fast rate. Considering this issue systems have contributed in summarization of these messages specially tweets. The previous system main focus was on summarization of all the messages and obtaining topic from these sets. Few system which contributed to tweet clustering and text summarization are explained in the below discussion.

Clustering the tweet streams:
In this system, the main focus was to cluster the coming tweets based on their similarity. The system formed initial clusters and then as the tweets stream arrived the similarity of the particular arrived tweet was obtained with each and every cluster and if the match found the tweet would then belong to that particular cluster. If in case the new arrived tweet does not match any of the cluster a new cluster will be formed for that newly arrived tweet. This system faced few problems when the tweets which arrived were of very different meanings then the system would form large amount of clusters which would again in turn increase user's efforts and reduce readability [4].

Clustering tweets stream with outlier analysis:
The major problem with the previous system was that the number of clusters. Few clusters would be created and would fall apart as no new tweets arrived which matched with that clusters. As a result system would form old ageing clusters. To overcome this problem new system was developed which for the clusters initially but also would perform outlier analysis for clusters and tweets which made sure that no clusters are formed for longer time even if no tweets are matching them. This made user life little easier [3].

Generating Summary of tweets:
This system focused on the generating the summary view of the tweets which arrived at a very high rate. The problem which user faced for readability was somewhat targeted and solved by providing the summary view of the tweets set. Summary included eliminating the repeated words and duplicate meaning tweets [1].

Summarization and timeline generation:
The major problem which the previous system faced where the rate at which the new tweets arrived. Also the previous system did not focused on the large collection of dynamically arriving tweets. This system used a SUMBLR framework which included tweets vector clustering and incremental clustering on continuous arriving tweets and also summary generation based on timeline and topic obtaining from the set. This system mainly focused on the dynamically arriving tweets and outlier analysis for the same. The outlier analysis also made sure that the old tweets were deleted which reduced the storage space required [5].

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

III. SYSTEM ARCHITECTURE

Table 1: Literature Review on different clustering and summarization method.

PAPER	AUTHOR	METHODS	ADVANTAGES	DRAWBACK/ LIMITATIONS
BIRCH: An Efficient Data Clustering Method for large database(1996)	T. Zhang, R. Ramkrishnan	BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies	Hierachical data structure is used for clustering , which made without scanning all data and currently existing clusters	Data space is not uniformly occupied
Clustering Data Stream(2003)	Nina Mishra, Adam Meyerson, Sudipto Guha	STREAM algorithm	Clustering can takes place in small space using divide and conquer algorithm	STREAM algorithm scan recursively number of times
A Framework For Clustering evolving Data Stream(2003)	C.C. Aggarwal, J. Wang, P. S. Yu	CluStream Algorithm	This algorithm divided into statistical data collection based on pyramidal time window	All these techniques fails to provide effective clustering
On Clustering Massive Text And Categorical Data Stream(2010)	C.C Aggarwal, P.S. Yu	Algorithms for text and categorical data stream clustering	Categorical data stream provides online analytical processing approach to stream clustering	Unable to summarize clustering Stream data
Multi-document Summarization via sentence level semantic analysis and symmetric matrix factorization(2008)	D. Wang, T. Li, S. Zhu	Algorithm for multi document summarization	Generate score for each word in set of document using machine learning	It have to maintain two data components for finding set of sentences from document cluster for maximize score
A Participant Based Approach For Event summarization Using Twitter Streams(2013)	C. Shen, F. Liu, F. Weng, T. Li	Participant based event summarization approach	Different types of events are considered and the participants take part into event for summarization	It provides summary to related participant of selected events only

This framework consists of five major modules which evaluates system according to user preference which are namely:

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

1) Tweet Stream Clustering Module
2) TCV Summarization Module
3) Topic Evolution Detection Module
4) Sentiment Analysis Module
5) Topic Popularity Module

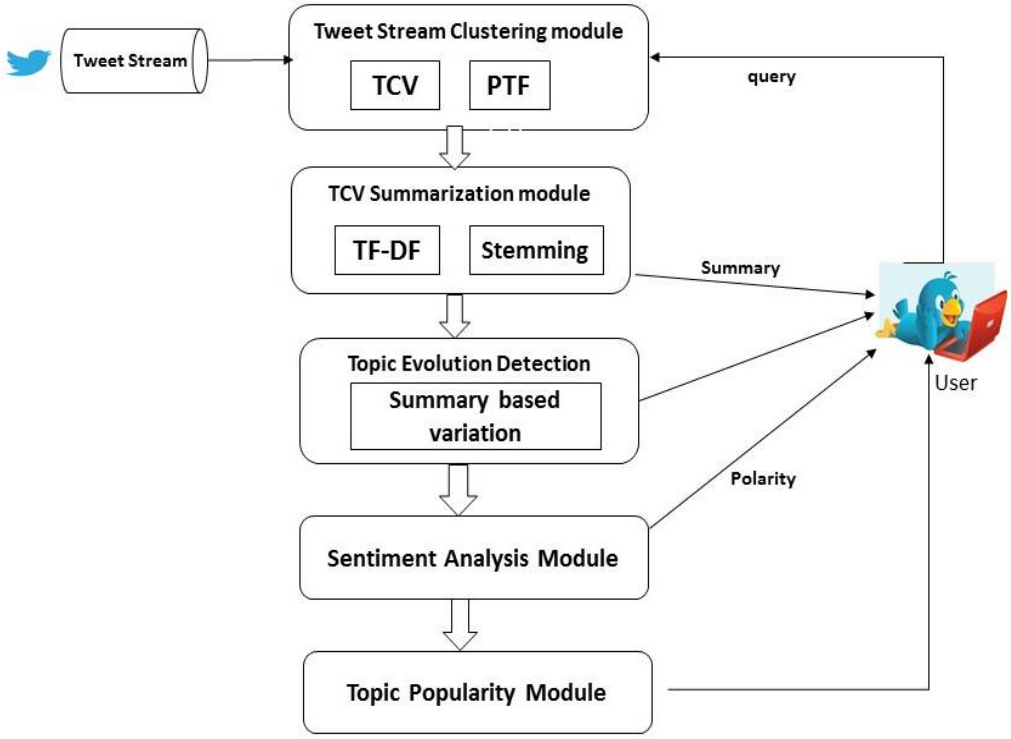

Figure 1: System Architecture.

Tweet Stream Clustering Module

The tweet stream clustering module maintains the given a topic-based tweet stream, it is able to efficiently cluster the tweets and maintain compact cluster information [2].

1. Initialization: At the start of the stream, a small number of tweets are collected and use a k-means clustering algorithm to create the initial clusters. The corresponding TCVs are initialized according to TCV concept. Next, the stream clustering process starts to incrementally update the TCVs whenever a new tweet arrives.
2. Increment clusters: Suppose a tweet t arrives at time t_s, and there are N active clusters at that time. The key problem is to decide whether to absorb t into one of the current clusters or upgrade t as a new cluster. We first find the cluster whose centroid is the closest to t.

   The minimum bounding similarity (MBS) measures the average closeness between the centroid and the tweets included in the cluster Cp. MBS is used to decide whether t is close enough to Cp: if MaxSim(t) is smaller than it, then t is upgraded to a new cluster; Otherwise, t is added to Cp.

   The MBS is defined as $^{\beta}.\overline{Sim(c_p)}$, where $^{\beta}$ is a bounding factor (0 < b < 1) and $^{\beta}.\overline{Sim(c_p)}$ is the average cosine similarity between the centroid of Cp and the tweets included in Cp. $\overline{Sim(c_p)}$ can be calculated as follows can be calculated as follows:

$$\overline{Sim(c_p)} = \frac{1}{n}\sum_{i=1}^{n}\frac{tvi.cv}{||tvi||.\ ||cv||} = \frac{cv}{n.\ ||cv||}\sum_{i=1}^{n}\frac{tvi}{||tvi||}$$

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

$$= \frac{wsum_v/n}{n.\ ||\frac{wsum_v}{n}||}\ .sum\_v\ = \frac{wsum_v\ .\ sum\_v}{n\ .\ ||wsum_v||}$$

3. Deleting Outdated Clusters: For most events (such as news, football matches and concerts) in tweet streams, timeliness is important because they usually do not last for a long time. Therefore it is safe to delete the clusters representing these sub - topics when they are rarely discussed. To find out such clusters, an intuitive way is to estimate the average arrival time (denoted as Avgp) of the last p percent of tweets in a cluster. However, storing p percent of tweets for every cluster will increase memory costs, especially when clusters grow big. Thus, we employ an approximate method to get Avgp.
4. Merging Clusters: If the number of clusters keeps increasing with few deletions, system memory will be exhausted. To avoid this, we specify an upper limit for the number of clusters as Nmax. When the limit is reached, a merging process starts. The process merges clusters in a greedy way. First, we sort all cluster pairs by their centroid similarities in a descending order. Then, starting with the most similar pair, we try to merge two clusters in it. When both clusters are single clusters which have not been merged with other clusters, they are merged into a new composite cluster.

TCV Summarization Module:

The TCV summarization module provides two types of summaries: online and historical summaries. An online summary describes what is currently discussed among the public. Thus, the input for generating online summaries is retrieved directly from the current clusters maintained in memory. On the other hand, a historical summary helps people understand the main happenings during a specific period, which means we need to eliminate the influence of tweet contents from the outside of that period. It uses different terms like TF-IDF and Stemming to generate summary [6].

Topic Evolution Detection Module:

Topic evolution detection algorithm which produces real-time and range timelines in a similar way. The algorithm discovers sub-topic changes by monitoring quantified variations during the course of stream processing. A large variation at a particular moment implies a sub-topic change, which is a new node on the timeline.

Sentiment Analysis Module:

Sentiment Analysis Module detects the polarity of the particular tweet. That is particular tweet is positive, negative, or neutral. It helps user that by observing only polarity view user may decide to take part in this event or not. In this polarity is calculated by using WordNet. WordNet automatically calculate polarity of each word, then average of words in tweet gives the polarity of that tweet.

Topic Popularity Module:

Topic popularity gives the ranking i.e. most popular tweets. The popularity of tweet is calculated using if the particular topic discussed more time then the topic is popular. Most ranking i.e. most popular occurs first and so on.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

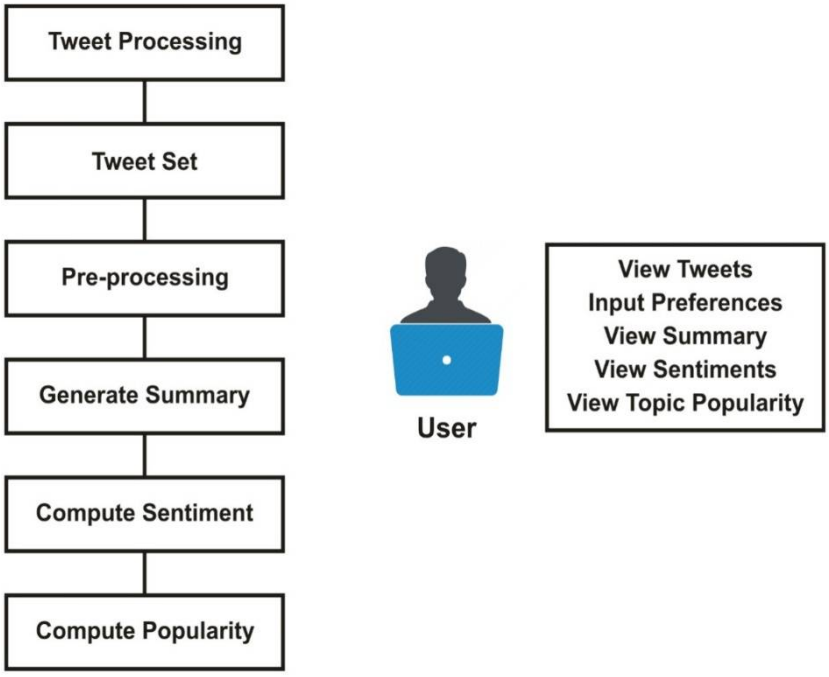

Figure 2: System workflow diagram.

IV. ALGORITHMS

System uses following algorithms:

4.1 Text Mining Algorithms:

4.1.1 Stemming

The stemming process finds the root words of any input words. It eliminates the tense part from the word. The stemming algorithm increases the comparison efficiency. We have used the standard porter stemmer [7] algorithm for the same. This algorithm works in five steps which are as follows.

Step 1: gets rid of plurals and -ed or –ing

Step 2: turns terminal y to i when there is another vowel in the stem.

Step 3: maps double suffices to single ones. so–ization.

Step 4: deals with -ic-, -full, -ness etc. Step 5: takes

off -ant, -ence etc.

1. Term Frequency: By using term frequency we can find out the occurrence of the word in that particular text data. To find term frequency [8] we have used the following equation:
   Term Frequency (TF) =occurrence of word/ total word count in the text data.
2. InverseDocumentFrequency: By using inverse document frequency we can find out the occurrence of the word all the rest of the text data in the dataset. To find inverse document frequency we have used the following equation:
   Inverse Document Frequency (IDF) =word occurrence in other document/ total number of text document
   Term Frequency Inverse Document Frequency (TF-IDF): By using this we get how important the word is to the document. To obtain TF-IDF we use the following equation.
   TF-IDF= TF of the word * IDF of the word.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

3. CosineEquations: The cosine equations are used to find difference between two numeric values. The following is the equation of cosine similarity:
   coeff_Cosine= x*y/Sqrt(x^2*y^2)

4.2.2 Tweet Stream Clustering Algorithm

Input: a cluster set C_set
Step 1: while *!stream.end()* do
Step 2: Tweet t = stream.next();
Step 3: *choose Cp in C set* whose centroid is the closest to *t*;
Step 4: if *MaxSim(t)* < *MBS* then
Step 5: create a new cluster C_{new} = {*t*};
Step 6: C_set.add(C_{new});
Step 7: else
Step 8: update *Cp* with *t*;
Step 9: if $TS_{current}\%(\alpha^i) == 0$ then
Step 10: store *C_set* into PTF;

4.2.3 Topic Evolution Detection

Input: a tweet stream binned by time units
Output: a timeline node set TN
Step 1: TN = ϕ;
Step 2: while *!stream.end()* do
Step 3: *Bin Ci = stream.next()*;
Step 4: if *hasLargeVariation()* then
Step 5: *TN.add(i)*;
Step 6: return *TN*;

V. EXPERIMENTAL SETUP AND RESULTS

The experiments are done on "Chelsea2015" dataset. The input to the system is Chelsea2015 dataset. It contains metadata such as user id, tweet, date and time of posted tweet, if there are retweet on any tweet, etc. And it downloaded from the URL: www.researchdatasets.com. It having nearby 2 lacs tweets record.

Our objective is to detect nodes in the reference timeline as the stream proceeds. We analyse result performance of the topic evolution detection algorithm using summary-based variations. We present precision, recall, and F-score of the timeline nodes detected by summary-based variation method. Since similar trends are observed in all Chelsea2015 data set.

Some basic result graphs shows below such as scalability on data size i.e. number of tweets vs running time (sec) and number of tweets vs memory utilised (KB).

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

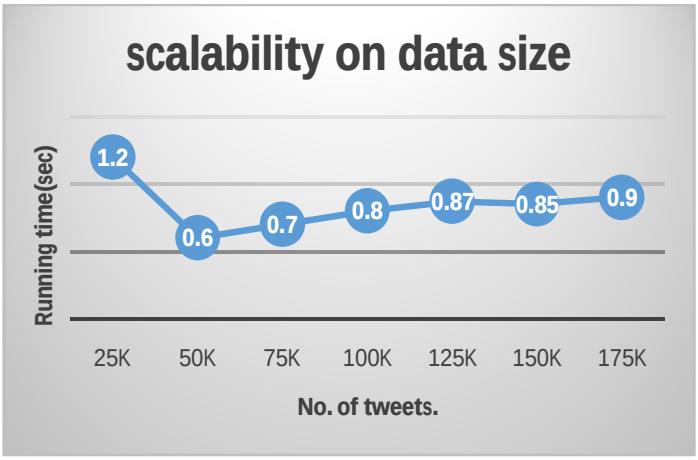

Figure 3: Graph for number of tweets vs running time (sec)

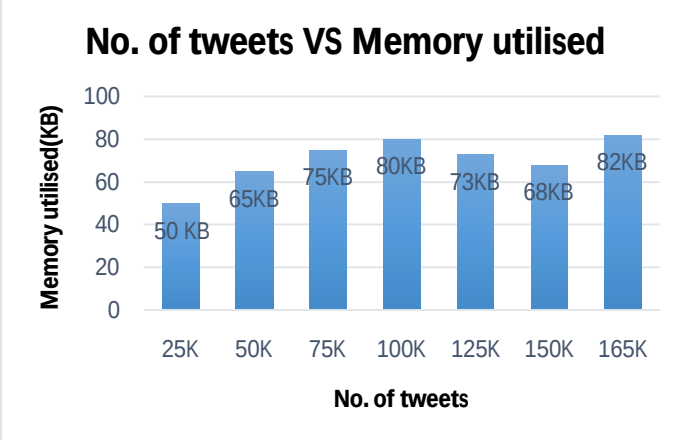

Figure 4: Graph for number of tweets vs memory utilized (KB).

The effect of decision threshold for SUM (τ_s) is shown in below figure as the threshold increases, recall declines while precision increases. This is expected since higher threshold would exclude more promising candidate nodes, and those remaining nodes with larger variations are more likely to be the correct ones. We get the highest F-scores when $\tau_{s = 1.1}$ for SUM and SUM has relatively higher recall. This is also consistent with our analysis SUM detects sub-topic changes based on content variation, so consider SUM as 'sensitive' method.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

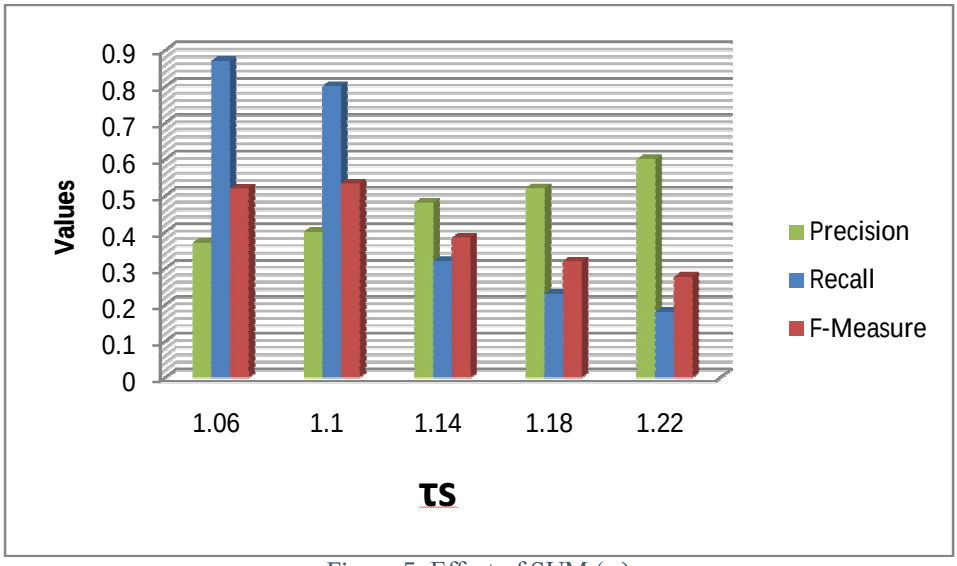

Figure 5: Effect of SUM (τ_s)

Following are formulae for calculating precision, recall and F-measure:

- Precision= $\frac{number\ of\ relevant\ tweet\ \cap\ number\ of\ retrieved\ tweet}{number\ of\ retrieved\ tweet}$
- Recall= $\frac{number\ of\ relevant\ tweet\ \cap\ number\ of\ retrieved\ tweet}{number\ of\ relevant\ tweet}$
- *F-measure*= $\frac{2\ \times\ precision\ value\ \times\ recall\ value}{precision\ value\ +\ recall\ value}$

We evaluate the proposed system method on Chelsea2015 data set and achieve effective clustering, summarization, topic evolution detection, sentiment view and topic popularity. The experimental results are done on system having configuration processor I5, RAM 4GB, and operating system windows 10. Results may vary if runs on different machine having different configurations.

VI. CONCLUSION AND FUTURE WORK

In this project, we proposed a continuous tweet stream summarization framework to generated summaries and timelines in the context of streams. It employs a tweet stream clustering algorithm to compress tweets into TCVs. Our proposed agglomerative clustering algorithm produced effective clusters, especially if the clusters are globular. We designed a novel data structure called TCV for stream processing, and proposed the TCV-Summarization algorithm for generating summary. The topic evolution can be detected automatically, allowing system to produce dynamic timelines for tweet streams. Polarity of tweets can be recognized that tweet is positive, negative or neutral. Topic popularity gives the ranking tweet by calculating distance measure of variations.

Intended System is made to work on only desktop applications in future we will extend its use for smart phone applications. The system currently does not consider emoticons to evaluate polarity of tweets in future we will make use of emoticons which will boost the accuracy.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 12, December 2017

REFERENCES

1. Zhenhua Wang, Lidan Shou, Ke Chan, G. Chen, and Sharad Mehrotra, "On Summarization and Timeline Generation for Evolutionary Tweet Streams," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 27, pp. 1301-1315, May 2015.
2. Nina Mishra, A. Meyerson, R. Motwani, Guha, and L. O'callaghan, "Clustering data stream: Theory and Practice," IEEE Transaction On Knowledge And Data Engineering, vol. 15, no. 3, pp. 515-528, June 2003.
3. T. Zhang, R. Ramkrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in Proc. ACM SIGMOD Int. Conf. Manage. Data, pp. 103-114, 1996.
4. C. C. Aggarwal, J. Wang, J. Han, and P. S. Yu, "A Framework For clustering evolving data streams," in Proc. 29th Int. Conf. Very large Data Bases, pp. 81-92, 2003.
5. D. Inouye, and J. K. Kalita, "Comparing twitter summarization algorithms for multiple post summaries," in Proc. IEEE 3rd Int. Conf. Social Comput., vol. 2, pp. 298-306, 2011.
6. C. C. Aggrarwal and P. S. Yu, "On clustering massive text and Categorical data streams," Knowl. Inf. Syst., vol. 24, pp. 171-196, 2010.
7. D. Wang, T. Li, S. Zhu, and C. Wing, "Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization," in Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, pp. 307-314, 2008.
8. D. R. Radev, and G. Erkan "LexRank: Graph-based lexical centrality as salience in text summarization," J. Artif. Int. Res., vol. 22, pp. 457-479, 2004. Mr. Rajesh H. Davda1, Mr. Noor Mohammed, " Text Detection, Removal and Region Filling Using Image Inpainting", International Journal of Futuristic Science Engineering and Technology, vol. 1 Issue 2, ISSN 2320 – 4486, 2013
9. Z. He, C. Chen, J. Bu, C. Wang, L. Zhang, X. He, "Document summarization based on data reconstruction," in Proc. 26th AAAI Conf. Artif. Intell., pp. 620-626, 2012.
10. J. Xu, D. V. Kalashnikov, and S. Mehrotra, "Efficient summarization framework for multi-attribute uncertain data," in Proc. ACM SIGMOD Int. Conf. Manage., pp. 421-432, 2014.
11. 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, pp. 1152-1162, 2013.