ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 6, June 2016

An Advanced approach for Summarization and Timeline Generation of Evolutionary Tweet Streams

P.Sindhuja, J. Suneetha,

ABSTRACT: **Tweet are being created short text message 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. Our proposed work consists three components tweet stream clustering to cluster using tweet cluster algorithm and second tweet cluster vector technique to generate rank summarization using greedy algorithm, therefore requires functionality which significantly differ from traditional summarization. In general, tweet summarization and third to detect and monitors the summary-based and volume based variation to produce timeline automatically from tweet stream. Implementing continuous tweet stream reducing a text document is however not an simple task, since a huge number of tweets are worthless, unrelated and raucous in nature, due to the social nature of tweeting. Further, tweets are strongly correlated with their posted instance and up-to-the minute tweets tend to arrive at a very fast rate. Efficiency—tweet streams are always very big in level, hence the summarization algorithm should be greatly capable; Flexibility—it should provide tweet summaries of random moment durations. Topic evolution—it should routinely detect sub-topic changes and the moments that they happen.**

Keywords: ***Tweet stream, continuous summarization, timeline, summary***

I. INTRODUCTION

The micro blogging site started in 2006 has become great popularity such as Twitter, facebook etc. This is resulted in explosion of the amount of short text messages. In February 2011, Twitter had 200 million registered users and 25 billion tweets in all of 2010. In this majority of post most of conversational or not meaningful, about 3.6% of the posts concern topics of mainstream news. Tweets, in their raw form, while being informative, can also be immense. The searching for a hot topic may yield millions of tweets, spanning weeks. The user unnecessarily goes through the millions of tweets and it is impossible every time. For this there is one solution namely filtering. Even if filtering is allowed, plowing for important contents, through such large amount of tweets is also very difficult and hard to possible task. This is happen because of enormous amount of irrelevant tweets. Another possible solution for information overload problem is summarization. The summarization is used to help what exactly the contents are conveying.

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. This technology makes use of a coherent summary such as length, style of writing and syntax. 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. Here are some examples of search engines in which summarization techniques are used such as Twitter, Facebook, and Google etc. Other category includes document summarization, image collection summarization and video summarization. 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, tries to automatically create a representative summary or abstract of the entire document, by finding the most informative sentences. Similarly, in image summarization the system finds the most representative and important (or salient) images. For tweet summarization mostly document summarization technique is used.

Two types of automatic summarization approaches: extraction and abstraction. The extractive summarization identifies relevant sentences that belong to the summary. In extraction based summarization task, the automatic system extracts objects from the entire collection, without modifying the objects itself. Examples of this include key phrase extraction, where the goal is to select individual words or phrases to "tag" a document and The goal of document summarization is to select whole sentences (without modifying them) to create a short paragraph summary. Similarly, in image collection summarization, the system extracts images from the collection without modifying the images themselves. On the other hand, abstraction based summarization task, involves paraphrasing sections of the source document. In general, abstraction can condense a text more strongly than extraction, but the programs which can do this are harder to develop as they require the use of natural language generation technology, which itself is a growing field. The grouping of similar tweets forms different clusters. These clusters used for summarization of tweet streams. Summarizing is defined as reduces the size of contents and indicate which particular topic is discussed on social sites. Top tweets are found out from clusters by using ranking algorithm.

Traditional document summarization techniques are not effective for big size tweets as well as not suitably applicable for tweets which are arrived fast and continuously. To overtake this problem tweet summarization is requires which should have new functionality significantly different from traditional.

summarization. Tweet summarization has to take into consideration the temporal feature of the arriving tweets. Consider example of Apple tweets [10]. A tweet summarization system will monitor Apple related tweets which are produced a real-time timeline of the tweet stream. Given a timeline range, the document system may generate a series of current time summaries to highlight points where the topic or subtopics evolved in the stream. Such a system will effectively enable the user to learn major news or discussion related to Apple without having to read through the entire tweet stream.

II. LITERATURE REVIEW

Tweet summarization includes two steps. First step requires tweet data clustering and then actually summarization is performed.

Algorithm for stream data clustering has been widely studied by various authors in literature. BIRCH is the balanced iterative reducing and clustering using hierarchies' algorithm. This algorithm is an unsupervised data mining algorithm [6]. It is used to perform hierarchical clustering over large data-sets. An advantage of BIRCH is that, it has ability to make clusters in increment and dynamic manner. This algorithm handles noise effectively and suitable for large databases. It makes cluster of incoming and multi-dimensional metric data points, to produce the best quality clustering for a given set of resources such as memory and time constraints. Bradley proposed a scalable clustering approach, which stores only important portions of the data with compressing or discarding other portions of the data which is not useful. This framework of clustering is based on the concept that effective clustering solutions is obtained by selectively storing important portions of the data and summarizing other portions of the data. The size of prespecified memory buffer which is allowable determines the amount of summarizing and required internal book-keeping. Author assumes that an interface to the database allows the algorithm to load number of data points requested. Data compression represents group of points by sufficient statistics. The interface to database allows the algorithm to load number of data points. These are obtained from a sequential scan, a random sampling or any means provided by the database engine [7]. CluStream is one of the most typical stream clustering methods. It having online micro-clustering component and also offline macro-clustering component. Online micro-clustering component require efficient process to store summaries. Offline components use only summary statistics. The pyramidal time frame was also proposed by authors to recall historical micro clusters for different time durations [5].

Here are some document summarization approaches are explained. Random Summarizer is an approach which randomly selects k posts or each topic as summary. This method was useful in order to provide worst case performance and also set the lower bound of performance [3]. Most Recent Summarizer approach chooses the most recent k posts as a summary from the selection pool. It is able to choose the first part of a news article as summary. This approach is implemented because the intelligent summarizers cannot perform better than simple summarizer. This summarizer only uses the first part of the document as summary [3]. LexRank summarizer uses a graph based method. It detects pairwise similarity between two sentences or between two posts. It makes the similarity score that is the weight of the edge between the two sentences. The final score of a post is computed based on the weights of the edges that are connected to each other. This summarizer is helpful to provide summarization based on baseline for graph instead of direct frequency summarization. Though it does depend on frequency, this system uses the relationships among sentences to add more information. LexRank Algorithm provides better view of important sentences. This is more complex algorithm than frequency based algorithm [1]. TextRank summarizer [2] is another graph based method. This approach uses the PageRank algorithm. This provided another graph based summarizer which incorporates potentially more information than LexRank. This is happens because it recursively changes the weights of posts. The final score of each post is dependent on how it is related to immediately connected posts as well as the way in which posts are related to other posts. TextRank algorithm is graph based approach used to find top ranked sentences. TextRank includes the whole complexity of the graph rather than just pair wise similarities. ETS (Evolutionary Timeline Summarization) [9] generates timelines for large amount of data. ETS gives evolutionary trajectories on particular dates. The advantage is that it facilitates fast news browsing. SPUR that is Summarization via Pattern Utility and Ranking is a novel algorithm used to summarize a batch of transactions with low compression ratio and high quality. It is working in a high scalable fashion. Xintian Yang et al. also develop D-SPUR which is the dynamic version of SPUR. D-SPUR is enhanced and modifies the pyramidal time window in data streams. SPUR and D-PUR algorithm compress messages with low compression ratio, high quality and fast running time [4]. Twitter streams also used for event summarization to represent information in live manner. The participant based approach is used for event summarization. The key components used for summarization are Participant Detection, Sub-event Detection and Summary Tweet Extraction. Participant detection identifies event participants then identify sub-events related to participants. The tweets are extracted from sub-events using Summary Tweet Extraction component [8]. Zhenhua Wang et al. introduce a summarization framework called Sumblr. This is the continuous summarization by stream clustering. Continuous summarization is difficult task as it contains large number of meaningless and irrelevant tweets. This is the first which

studied continuous tweet stream summarization. This framework consists of three main components, namely the Tweet Stream Clustering module, the High-level Summarization module and the Timeline Generation module. Sumblr is useful to work on dynamic, fast arriving, and large-scale tweet streams [10].

III. PROPOSED WORK

This proposed appraoch, we introduce a novel summarization framework called Sumblr (continuouS sUMmarization By stream cLusteRing). The framework consists of three main components, namely the Tweet Stream Clustering module, the High-level Summarization module and the Timeline Generation module.In the tweet stream clustering module, we design an efficient tweet stream clustering algorithm, an online algorithm allowing for effective clustering of tweets with only one pass over the data.The high-level summarization module supports generation of two kinds of summaries: online and historical summaries.The core of the timeline generation module is a topic evolution detection algorithm, which consumes online/historical summaries to produce real-time/range timelines. The algorithm monitors quantified variation during the course of stream processing.

III. SYSTEM ARCHITECTURE

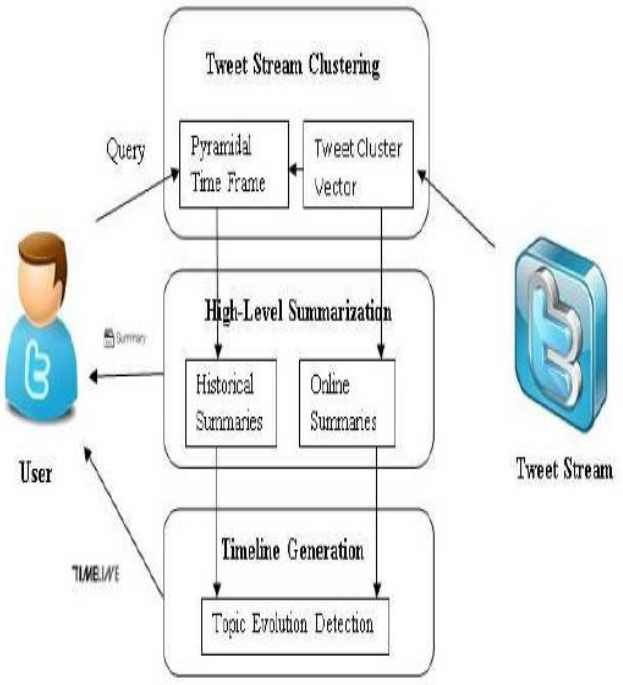

Fig. 1 System Architecture

The framework consists of three main components, namely

- The Tweet Stream Clustering module,
- The High-level Summarization module
- Timeline Generation module.

Tweet stream clustering module : we design an efficient tweet stream clustering algorithm, an online algorithm allowing for effective clustering of tweets with only one pass over the data. This algorithm employs two data structures to keep important tweet information in clusters. The first one is a novel compressed structure called the tweet cluster vector (TCV). TCVs are considered as potential sub-topic delegates and proyect dynamically in memory during stream processing. The second structure is the pyramidal time frame (PTF) [1], which is used to store and organize cluster snapshots at different moments, thus allowing historical tweet data to be retrieved by any arbitrary time durations.

The high-level summarization module It supports generation of two kinds of summaries: online and historical summaries. (1) To generate online summaries, we propose a TCV-Rank summarization algorithm by referring to the current clusters maintained in memory. This algorithm first computes centrality scores for tweets kept in TCVs, and selects the top-ranked ones in terms of content coverage and novelty. (2) To compute a historical summary where the user specifies an arbitrary time duration, we first retrieve two historical cluster snapshots from the PTF with respect to the two endpoints (the beginning and ending points) of the duration. Then, based on the difference between the two cluster snapshots, the TCV-Rank summarization algorithm is applied to generate summaries.

The Timeline generation module The timeline generation module : is a topic evolution detection algorithm, which consumes online/historical summaries to produce real-time/range timelines. The algorithm monitors quantified variation during the course of stream processing. A large variation at a particular moment implies a sub-topic change, leading to the addition of a new node on the timeline. In our design, we consider three different factors respectively in the algorithm. First, we consider variation in the main contents discussed in tweets (in the form of summary). To quantify the summary based variation (SUM), we use the Jensen-Shannon divergence (JSD) to measure the distance between two word distributions in two successive summaries. Second, we monitor the volume-based variation (VOL) which reflects the significance of sub-topic changes, to discover rapid increases (or “spikes”) in the volume of tweets over time. Third, we define the sum-vol variation (SV) by combining both effects of summary content and significance, and detect topic evolution whenever there is a burst in the unified variation.

V. CONCLUSION

Here we studied various approaches for document summarization such as filtering and tweet summarization. These approaches are used for managing huge amount of tweets. Filtering is not an efficient approach because of tweet data is noisy and redundant. Because of the summarization is used to summarize the tweet data. Traditional document summarization techniques are not effective for big size tweets as well as not suitably applicable for tweets which are arrived fast and continuously, also they are not focus on static and small-scale data set. To overcome this problem, develop a multi topic version of a continuous tweet stream

summarization framework, namely Sumbler to generate summaries and timelines in the context of streams, which will also suitable in distributed systems and evaluate it on more complete and large scale data sets, which deals with dynamic, fast arriving, and large- scale tweet streams. This will discovers the changing dates and timelines dynamically during the process of continuous summarization. Moreover ETS (Evolutionary Timeline Summarization) does not focus on efficiency and scalability issues which are very important in our streaming context.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] L. Gong, J. Zeng, and S. Zhang, "Text stream clustering algorithm based on adaptive feature selection," Expert Syst. Appl., vol. 38, no. 3, pp. 1393–1399, 2011.

[5] Q. He, K. Chang, E.-P. Lim, and J. Zhang, "Bursty feature representation for clustering text streams," in Proc. SIAM Int. Conf. Data Mining, 2007, pp. 491–496.

[6] J. Zhang, Z. Ghahramani, and Y. Yang, "A probabilistic model for online document clustering with application to novelty detection," in Proc. Adv. Neural Inf. Process. Syst., 2004, pp. 1617–1624.

[7] S. Zhong, "Efficient streaming text clustering," Neural Netw., vol. 18, nos. 5/6, pp. 790–798, 2005.

[8] 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.

[9] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," in Proc. ACL Workshop Intell. Scalable Text Summarization, 1997, pp. 10–17.

[10] W.-T. Yih, J. Goodman, L. Vanderwende, and H. Suzuki, "Multidocument summarization by maximizing informative contentwords," in Proc. 20th Int. Joint Conf. Artif. Intell., 2007, pp. 1776–1782.

[11] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," J. Artif. Int. Res., vol. 22, no. 1, pp. 457–479, 2004.

[12] D. Wang, T. Li, S. Zhu, and C. Ding, "Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization," in Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2008, pp. 307–314.

[13] Z. He, C. Chen, J. Bu, C. Wang, L. Zhang, D. Cai, and X. He, "Document summarization based on data reconstruction," in Proc. 26th AAAI Conf. Artif. Intell., 2012, pp. 620–626.

[14] J. Xu, D. V. Kalashnikov, and S. Mehrotra, "Efficient summarization framework for multi-attribute uncertain data," in Proc. ACM SIGMOD Int. Conf. Manage., 2014, pp. 421–432.

[15] 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.

[16] 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.

[17] S. M. Harabagiu and A. Hickl, "Relevance modeling for microblog summarization," in Proc. 5th Int. Conf. Weblogs Social Media, 2011, pp. 514–517.

[18] H. Takamura, H. Yokono, and M. Okumura, "Summarizing a document stream," in Proc. 33rd Eur. Conf. Adv. Inf. Retrieval, 2011, pp. 177–188.

[19] 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.

[20] D. Chakrabarti and K. Punera, "Event summarization using tweets," in Proc. 5th Int. Conf. Weblogs Social Media, 2011, pp. 66–73. [21] M. Kubo, R. Sasano, H. Takamura, and M. Okumura, "Generating live sports updates from twitter by finding good reporters," in Proc. IEEE Int. Joint Conf. Web Intell. Agent Technol., 2013, pp. 527–534.

[22] 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.

[23] 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.

[24] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang, "Evolutionary timeline summarization: A balanced optimization framework via iterative substitution," in Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2011, pp. 745–754.

[25] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Twitinfo: Aggregating and visualizing microblogs for event exploration," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2011, pp. 227–236.

[26] J. Nichols, J. Mahmud, and C. Drews, "Summarizing sporting events using twitter," in Proc. ACM Int. Conf. Intell. User Interfaces, 2012, pp. 189–198.

[27] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis, "Discovering geographical topics in the twitter stream," in Proc. 21st Int. Conf. World Wide Web, 2012, pp. 769–778.

[28] C. Lin, C. Lin, J. Li, D. Wang, Y. Chen, and T. Li, "Generating event storylines from microblogs," in Proc. 21st ACM Int. Conf. Inf. Knowl. Manage., 2012, pp. 175–184.

[29] X. Yang, A. Ghoting, Y. Ruan, and S. Parthasarathy, "A framework for summarizing and analyzing twitter feeds,"

in Proc. Knowl. Discovery Data Mining, 2012, pp. 370–378.
[30] B. Van Durme, "Streaming analysis of discourse participants," in Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learning, 2012, pp. 48–58.
[31] C. Chen, F. Li, B. C. Ooi, and S. Wu, "TI: An efficient indexing mechanism for real-time search on tweets," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 649–660.
[32] D. Zwillinger, CRC Standard Mathematical Tables and Formulae. Boca Raton, FL, USA: CRC Press, 2011.
[33] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 1998, pp. 335–336.
[34] J. Lin, "Divergence measures based on the shannon entropy," IEEE Trans. Inf. Theory, vol. 37, no. 1, pp. 145–151, Jan. 1991.
[35] A. Zubiaga, D. Spina, E. Amigo, and J. Gonzalo, "Towards real-time summarization of scheduled events from twitter streams," in Proc. 23rd ACM Conf. Hypertext Social Media, 2012, pp. 319–320.
[36] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in Proc. ACL Workshop Text Summarization Branches Out, 2004, pp. 74–81.
[37] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in Proc. Human Lang. Technol. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2003, pp. 71–78.

P.Sindhuja, *MTech,*
Siddarth Institute of Engineering and Technology
J. Suneetha, *M.E.(,Ph.D.),*Associate Professor,
Siddarth Institute of Engineering and Technology