|| Volume 1 ||Issue 3 ||June 2016||ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

Tweet Streams Online Summarization and Timeline Generation

Miss. Geeta G. Dayalani

P.G Student, Computer Science & Engineering , Everest Educational Society's Group of Institutions, Aurangabad, Maharashtra, India.

***Abstract*—Twitter is one of the famous micro blogging services, with hundreds of millions of tweets being posted every day on a wide variety of topics. Tweets are the short text messages which are limited upto 140 characters in length. They are generated and given out at an unusual rate. Tweets are descriptive in nature. Tweets contain too many noisy data and redundancies. In this, we make an attempt to introduce a distinctive new framework for continuous summarization called Summblr to deal with the problem. Existing methods that summarize the documents high light on small scale data sets that are static. Summblr. in contrast ,is developed to deal with large-scale tweet data streams which arrive at a faster rate dynamically. Our constructed framework includes three major components. As a first step, a clustering algorithm for tweet data stream is proposed which is online to cluster together the tweets and maintain it in one of the data structure called as TCV that is tweet cluster vector. Secondarily, a novel technique called TCV-Rank summarization for generating summaries both online and historical of any time durations is proposed. Thirdly, we develop a method for effectively detecting the topic evolution, which continuously checks the variations that are summary or volume based to automatically produce the timelines from large tweet streams.**

Keywords:- Summarization, Timeline, Tweet stream, Specification, cluster.

I INTRODUCTION

The micro blogging site called twitter which started in the year 2006, has become a social remarkable happening. Over 400 million tweets are received by twitter daily. In the raw form tweets, while being descriptive, can also be overwhelming. If we search for any trendy topic in Twitter, it may result in millions of tweets, ranging weeks. Even if some criteria such as filtering is done, searching through large number of tweets for prominent data would be a illusion, by taking into account the huge volume of repetition or redundancy and noise that one might encounter. To make things more badly, novel tweets which satisfy the criteria of filtering may arrive at an incalculable rate continuously. There are number of solutions to the overload problem of data. One solution possible to overload problem of data is summarization. Summarization basically represents a collection of some documents with a summary which consists of number of sentences. A summary which is good should possibly cover all the important topics including the sub-topics and have diversification among various sentences to reduce the redundancy to a greater extend.

Summarization is used largely in content or data presentation, especially when users search on the internet by using their handheld devices like mobile or cell phones which have very smaller screens when compared to Personal computers. Traditional approaches of document summarization are not so effective in regard of tweets by taking into consideration both the huge number of tweets and also the fast and the endless nature of the tweet arrival. For this reason, Tweet summarization, requires features which are significantly different from traditional summarization functionalities. In addition, summarization of tweets has to take into account the temporal nature of the ample number of arriving tweets.

The tweet summarization system must hold the two queries as follows: summaries of random time durations and also the real time timelines. Such application would be helpful to easily navigate in relevant topic tweets and also support various other analysis tasks like instant reports or historical survey. In this paper, a new summarization method called endless summarization for tweet data streams is proposed.

To implement continuous or endless tweet stream summarization is however a very tough task, since ample of tweet data are not meaningful, and hence useless in nature. Furthermore, tweets are firmly related with their time that is posted and new tweets arrive at a very faster rate. Finally, a better solution for continuous tweet summarization has to pay heed on these three issues namely:

(1) Effectiveness (2) Elasticity(3) Topic evolution

Existing techniques of summarization fails to satisfy the above three requirements because:

(1) They focus on small data sets that are static, and hence are not effective for huge data sets and streams.
(2) Iterative summarization needs to be performed for each time duration, which is not feasible.

(3) Results obtained from summary are not sensitive to time. Thus it not easy for them to detect evolution in any topic.

In this paper, we make an attempt to propose a new framework for summarization called Summblr. Summblr is used for continuous summarization by stream clustering.

The main objective of this is as follows:

_A framework called Summblr that is summarization by clustering is proposed, to generate the summaries and timelines of tweet data streams.

_ Anew data structure is developed called TCV for processing of tweets, and propose the TCV Rank algorithm for both summarization online as well as historical.

_ A topic evolution identification algorithm which generates timelines by monitoring three types of variations is proposed

II. RELATEDWORKS

Here, we describe some of the related existing research work and discuss how our work differs from it.

Summarization of tweets is a two step process. In the first step, tweet data is clustered together and in the second step actual summarization is done.

A number of algorithms have been developed for document summarization during recent years. Notable algorithms include Sum Basic [5] and the centroid algorithm [6]. Sum Basic's imply that words that occur more frequently across documents have a greater chance of selecting for human created multidocument summaries than the words that occur less frequently in n number of documents. The centroid algorithm firstly takes a centrality measure of a sentence based on the overall topic of the whole document cluster or to a document in the case of single document summarization. The LexRank algorithm [7] is used for computing the importance of sentences or other textual data units in one document or set of the Text Rank algorithm [8] is a graph-based approach which tries to find the most highly ranked sentences or specifically keywords in a document using the Page Rank algorithm [9].

A. Stream Data Clustering

Stream data clustering has been studied in the literature to a greater extent. BIRCH [4] clusters the data by using an in-memory structure called CF-tree instead of the original or actual huge data set. Bradley et al. [1] proposed another framework of clustering which stores only the important parts of the data selectively, and discards other parts of data which are waste. CluStream [3] is one of the most important stream clustering methods. It contains an online micro-clustering content and an offline macro-clustering component. The authors also proposed pyramidal time frame [3] to recall historical data clusters of any particular time durations.

In [13], the authors extended CluStream algorithm to generate duration-based results of clustering for text data as well as categorical tweet data streams. However, the algorithm depends on an online-phase to produce a large number of "micro-clusters" and an off-line-phase to re-cluster them together. Our tweet stream clustering algorithm is one of the online method with no offline clustering required.

B. Microblogging and Twitter

There has been much recent interest on determining and then tracking the evolution of events on Twitter and other social media websites, e.g., discussions about an volcanic eruption or earthquake on twitter [10], detecting new events which are also called first stories in the tweet-stream [12], visualizing the evolution of tags [11] and other events on Flickr, YouTube, and Face book [4, 3]. The problem has also been approached from the point of view of efficiency: [13] propose indexing and compression techniques to speed up event detection without sacrificing detection accuracy.. We assume that the event detection has already been performed, possibly using one of the aforementioned techniques; our goal is to collate all the information in the tweets and present a summarized timeline of the event.

C. Document/Microblog Summarization

Summarization of multiple documents can also be performed by using two approaches: extractive and the second is abstractive. Extractive summarization is a process in which the sentences are selected from the number of documents itself, while abstractive summarization produces the phrases which may not be present in the actual existing document.

In this paper, we concentrate on extractive summarization. Salient scores are assigned to the sentences of the documents, and then the sentences are selected according to the ranking. Top-ranked sentences are selected initially [14], [15], [16]. Some authors try to retrieve the summaries without such salient scores.

Abstractive summarization is not easy on Twitter streams. It is easily affected by noise and redundancy or by the large variations of tweets.

III THE SUMMBLR FRAMEWORK

As shown in Figure 1, our framework basically comprises of three main modules namely: clustering of tweet stream module, the high-level specification module and the timeline creation module. In this section, we shall consider each of them in detail. In the first module that is clustering of tweet stream, we an effective clustering algorithm is designed. An algorithm that is online which allows for effectively clustering the tweets together with only a single through over the entire data. This algorithm uses two new data structures that are, the first one is called the tweet cluster vector (TCV). During the processing of tweet streams, TCVs are maintained dynamically in the memory. The second data structure is the pyramidal time frame (PTF) [3].PTF is basically used to save and organize the

snapshots of cluster at various different moments, thus ultimately allowing the historical data of tweets to be recovered by any arbitrary durations of time.

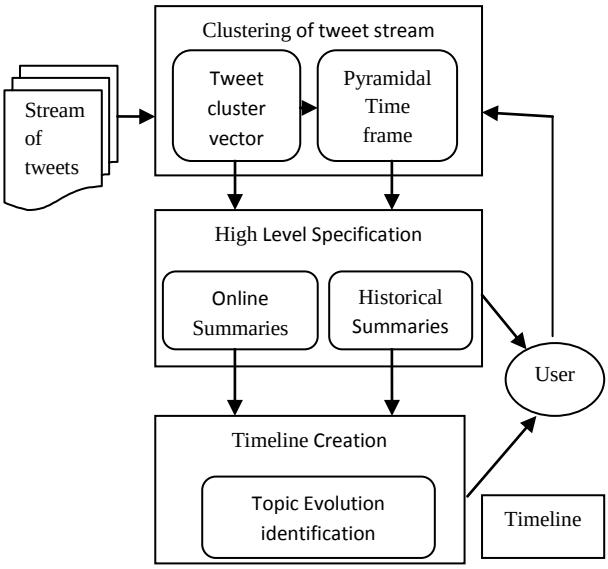

Figure 1 the framework of Summblr

The high-level summarization or specification module supports the creation of two types of summaries namely: online as well as historical summaries.

i) To produce the summaries online, an algorithm called TCV Rank summarization algorithm which takes into consideration current or active clusters that are stored dynamically in memory is proposed. ii) To generate a summary that is historical, the user mentions any particular timely duration. Initially, we take into picture two historical snapshots of cluster from the Pyramidal frame which represents the starting and ending point of the particular time duration. Then, based upon the change in snapshots, the TCV Rank summarization algorithm is eventually used to produce the summaries.

The timeline creation module is a topic evolution identification algorithm, which takes online or historical summaries as input to generate real-time or range timelines. This module basically discovers sub topic changes by monitoring the quantified variations during tweet stream processing.

IV CONCLUSION

We highlight on summarizing stream of tweets with respect to trendy topics along timelines to produce a survey of topic evolution which is defined by sub topics. A prototype called Summblr is proposed which supports tweet summarization continuously. Summblr makes use of a clustering algorithm of tweet stream to reduce the stream of tweets into TCVs structure and maintains them online dynamically. Thereafter, it uses a TCV Rank summarization algorithm for producing two types of summaries namely online as well as historical summaries with time durations arbitrarily. Lastly, the topic evolution can also be determined automatically, which allows the framework to create timelines dynamically for large number of tweet streams.

ACKNOWLEDGEMENT

It is my great pleasure in expressing sincere and deep gratitude towards my guide Prof. Rajesh. A. Auti. I am also thankful to Head of Department of Computer Science and Engineering, Prof. Rajesh. A. Auti for providing me various resources and infrastructure facilities. I also offer my most sincere thanks to Principal Prof. R. D. Shelke, Everest College of Engineering, Aurangabad, my colleagues and staff members of Computer Science and Engineering Department, Everest college of Engineering, Aurangabad for cooperation provided by them in many ways.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion," Information Processing & Management, vol. 43, no. 6, pp. 1606–1618, 2007.

[6] D. Radev, S. Blair-Goldensohn, and Z. Zhang, "Experiments in single and multi-document summarization using mead," DUC-01, vol. 1001, p. 48109, 2001.

[8] G. Erkan and D. Radev, "Lexrank: graph-based centrality as salience in text summarization," Journal of Artificial Intelligence Research, vol. 22, pp. 457–480, 2004.

[9] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in EMNLP. Barcelona: ACL, 2004, pp. 404–411.

[10] S. Brin and L. Page, "The anatomy of a large-scale hypertextualWebsearch engine* 1," Computer networks and ISDN systems, vol. 30, no. 1-7, pp. 107–117, 1998.

[11] 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.

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

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

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

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

[16] 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.

[17] 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.