International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 4, Issue 1 (January-February, 2016), PP. 168-170

ON SUMMARIZATION AND TIMELINE GENERATION FOR EVOLUTIONARY TWEET STREAMS

Kamran Ansari[1], Shamali Jadhav[2],Preeti Patole[3,] Kajal Patil[4]
Department, Of Computer Engineering,
Savitribai Phule Pune University,
Maharashtra, India
[1]iemkamran@gmail.com
[2]shamalijadhav24@gmail.com
[3]preeti10patole@gmail.com
[4]kajalpatil158@gmail.com

Abstract— **At an unprecedented rate, short-text messages such as tweets are being created and shared. While being informative, can also be overwhelming, tweets, in their raw form. It is a nightmare to plow through millions of tweets which contain enormous amount of noise and redundancy, for both end-users and data analysts. We propose a novel continuous summarization framework called Sumblr to alleviate the problem, in this paper. Sumblr is designed to deal with dynamic, fast arriving, and large-scale tweet streams, in contrast to the traditional document summarization methods which focus on static and small-scale data set. Our proposed framework consists of three major components. To cluster tweets and maintain distilled enumeration in a data structure called tweet cluster vector (TCV), we propose an online tweet stream clustering algorithm. We proposed a TCV-Rank summarization technique for generating online summaries and historical summaries of arbitrary time durations. Which monitors summary-based/volume-based variations to produce timelines automatically from tweet streams, we design an effective topic evolution detection method. Our experiments on large-scale real tweets demonstrate the efficiency and effectiveness of our framework.**

Index terms- **Tweet stream, continuous summarization, summary, timeline.**

I. INTRODUCTION

Rising popularity of microblogging services such as Twitter, Weibo, and Tumblr has resulted in the explosion of the amount of short-text messages. Twitter, for instance, which receives over 400 million tweets per day1 has emerged as an invaluable source of news, blogs, opinions, and more. While being informative, can also be overwhelming, tweets, in their raw form. Twitter may yield millions of tweets, spanning weeks, for instance, search for a hot topic. Plowing through so many tweets for important contents would be a nightmare, not to mention the enormous amount of noise and redundancy that one might encounter, even if filtering is allowed.

At an unpredictable rate, to make things worse, new tweets satisfying the filtering criteria may arrive continuously. To information overload problem is summarization, one possible solution. A good summary should cover the main topics (or subtopics) and have diversity among the sentences to reduce redundancy, summarization represents a set of information by a summary consisting of several sentences. Specially when users surf the internet with their mobile devices which have much smaller screens than PCs, summarization is extensively used in content presentation. However, are not as effective in the

context of tweets given both the large volume of tweets as well as the fast and continuous nature of their arrival, traditional document summarization approaches. Tweet summarization, therefore, requires functionalities which significantly differ from traditional summarization. In general, tweet summarization has to take into consideration the temporal feature of the arriving tweets. Since a big number of tweets are worthless, irrelevant and noisy in nature, due to the social nature of tweeting, implementing stable tweet stream summarization is however not an easy task. Using an illustrative example of a usage of such a system, Let us illustrate the desired properties of a tweet summarization system. for example, tweets about "Apple", consider a user interested in a topic-related tweet stream. A real-time timeline of the tweet stream, a tweet summarization system will constantly monitor "Apple" related tweets producing. A user may explore tweets based on a timeline. To highlight points where the topic/subtopics evolved in the stream, Given a timeline range, the summarization system may produce a sequence of time stamped summaries. To learn major news/discussion related to "Apple" without having to read through the entire tweet stream, such a system will effectively enable the user. a user may decide to zoom in to get a more detailed report for a smaller duration, given the big picture about topic evolution about "Apple". To get additional details for that duration, the system may provide a drill-down summary of the duration that enables the user. To obtain a roll-up summary of tweets, a user, perusing a drill-down summary, may alternatively zoom out to a coarser range. The summarization system must support the following two queries: summaries of arbitrary time durations and real-time/range timelines, to be able to support such drill-down and roll-up operations. But also support a range of data analysis tasks such as instant reports or historical survey, such application would not only facilitate easy navigation in topic-relevant tweets. To

this end, in this paper, we propose a new summarization method, continuous summarization, for tweet streams.

II. LETURATURE SURVAY

[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.

TCVs are considered as potential sub-topic delegates and maintained dynamically in memory during stream processing. The second structure is the pyramidal time frame (PTF), 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.

[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.

Clusters the data based on an in-memory structure called CF-tree instead of the original large data set. Bradley et al. [3] proposed a scalable clustering framework which selectively stores important portions of the data, and compresses or discards other portions.

[3] 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.

A variety of services on the Web such as news filtering, text crawling, and topic detecting etc. have posed requirements for text stream clustering. A few algorithms have been proposed to tackle the problem. Most of these techniques adopt partition-based approaches to enable online clustering of stream data. As a consequence, these techniques fail to provide effective analysis on clusters formed over different time durations.

III. PROPOSED SYSTEM

We propose a continuous tweet stream summarization framework, namely Sumblr, to generate summaries and timelines in the context of streams. We design a novel data structure called TCV for stream processing, and propose the TCV-Rank.algorithm for online and historical summarization. We propose a topic evolution detection algorithm which produces timelines by monitoring three kinds of variations.Extensive experiments on real Twitter data setsdemonstrate the efficiency and effectiveness of our framework.

IV. SYSTEM ARCHITECTURE

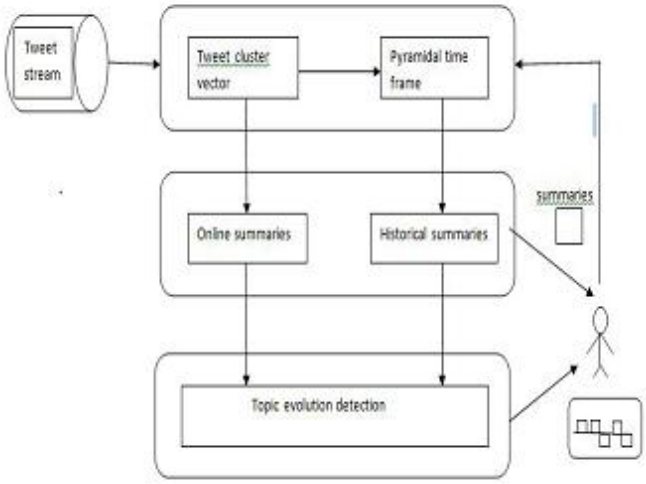

Fig.1. The Framework of Sumblr

we introduced a novel summarization structure called Sumblr (continuouS sUMmarization By stream cLusteRing). To the best of our knowledge, our work is the first to study continuous tweet stream summarization. The overall framework is depicted in Fig. 2. 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. 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 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 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. 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

We proposed a prototype called Sumblr which supported continuous tweet stream summarization. Sumblr employs a tweet stream clustering algorithm to compress tweets into TCVs and maintains them in an online fashion. Then, it uses a TCV-Rank summarization algorithm for generating online summaries and historical summaries with arbitrary time durations. The topic evolution can be detected automatically, allowing Sumblr to produce dynamic timelines for tweet streams. The experimental results demonstrate the efficiency and performance of our method. For future work, we aim to develop a multi-topic version of Sumblr in a distributed system, and estimate it on more complete and large-scale data sets.

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.