**INTRODUCTION**

INCREASING 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. Tweets, in their raw form, while being informative, can also be overwhelming. For instance, search for a hot topic in Twitter may yield millions of tweets, spanning weeks. Even if filtering is allowed, 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. 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 represents a set of documents by a summary consisting of several sentences. Intuitively, a good summary should cover the main topics (or subtopics) and have diversity among the sentences to reduce redundancy. Summarization is extensively used in content presentation, specially when users surf the internet with their mobile devices which have much smaller screens than PCs. Traditional document summarization approaches, 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. 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. Let us illustrate the desired properties of a tweet summarization system using an illustrative example of a usage of such a system. Consider a user interested in a topic-related tweet stream, for example, tweets about “Apple”. A tweet summarization system will continuously monitor “Apple” related tweets producing a real-time timeline of the tweet stream. As illustrated in in this system, a user may explore tweets based on a timeline (e.g., “Apple” tweets posted between October 22nd, 2012 to November 11th, 2012). Given a timeline range, the summarization system may produce a sequence of times tamped summaries to highlight points where the topic/subtopics evolved in the stream. Such a system will effectively enable the user to learn major news/ discussion related to “Apple” without having to read through the entire tweet stream.

Given the big picture about topic evolution about “Apple”, a user may decide to zoom in to get a more detailed report for a smaller duration (e.g., from 8 am to 11 pm on November 5th). The system may provide a drill-down summary of the duration that enables the user to get additional details for that duration. A user, perusing a drill-down summary, may alternatively zoom out to a coarser range (e.g., October 21st to October 30th) to obtain a roll-up summary of tweets. To be able to support such drill-down and roll-up operations, the summarization system must support the following two queries: summaries of arbitrary time durations and real-time/range timelines. Such application would not only facilitate easy navigation in topic-relevant tweets, but also support a range of data analysis tasks such as instant reports or historical survey.To this end, in this paper, we propose a new summarization method, continuous summarization, for tweet streams.

Implementing continuous tweet stream summarization is however 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, tweets are strongly correlated with their posted time and new tweets tend to arrive at a very fast rate. Consequently, a good solution for continuous summarization has to address the following three issues: (1) Efficiency—tweet streams are always very large in scale, hence the summarization algorithm should be highly efficient; (2) Flexibility—it should provide tweet summaries of arbitrary time durations. (3) Topic evolution—it should automatically detect sub-topic changes and the moments that they happen.

Unfortunately, existing summarization methods cannot satisfy the above three requirements because: (1) They mainly focus on static and small-sized data sets, and hence are not efficient and scalable for large data sets and data streams. (2) To provide summaries of arbitrary durations, they will have to perform iterative/recursive summarization for every possible time duration, which is unacceptable. (3) Their summary results are insensitive to time. Thus it is difficult for them to detect topic evolution . In this paper, we introduce a novel summarization framework called Sumblr (continuouS sUMmarization By stream cLusteRing). To the best of our knowledge, our work is the first to study continuous tweet stream summarization.