CLEar: A Real-time Online Observatory for Bursty and Viral Events

Runquan Xie [#+*], Feida Zhu [#], Hui Ma [#], Wei Xie [#], Chen Lin [+$]
[#] School of Information Systems, Singapore Management University, Singapore
[+] School of Information Science and Engineering, Xiamen University, Xiamen, China
[#] {rqxie, fdzhu, huima, wei.xie.2012}@smu.edu.sg
[$] chenlin@xmu.edu.cn

ABSTRACT

We describe our demonstration of CLEar (*Cl*airaudient *Ear*), a real-time online platform for detecting, monitoring, summarizing, contextualizing and visualizing bursty and viral events, those triggering a sudden surge of public interest and going viral on micro-blogging platforms. This task is challenging for existing methods as they either use complicated topic models to analyze topics in a off-line manner or define temporal structure of fixed granularity on the data stream for online topic learning, leaving them hardly scalable for real-time stream like that of Twitter. In this demonstration of CLEar, we present a three-stage system: First, we show a real-time bursty event detection module based on a data-sketch topic model which makes use of acceleration of certain stream quantities as the indicators of topic burstiness to trigger efficient topic inference. Second, we demonstrate popularity prediction for the detected bursty topics and event summarization based on clustering related topics detected in successive time periods. Third, we illustrate CLEar's module for contextualizing and visualizing the event evolution both along time-line and across other news media to offer an easier understanding of the events.

1. INTRODUCTION

Compared against traditional news media, social network services like Twitter have been recognized as much more responsive and reliable sources to pick up events that trigger a surge of public response within a short period of time, which we call *"bursty topics"*. In fact, in cases like natural disasters such as 2011 Japan earthquake, Twitter was virtually the lifeline of millions of people affected to obtain life-critical information on the time scale of minutes. Undoubtedly, the most valuable and unique feature of Twitter is its real-time responsiveness, appealing to a wide range of target users from businesses monitoring brand image, government sectors listening to social voices to organizations rendering relief for natural disasters. For a system using social media like Twitter to keep track of things happening around, one would be looking for the following traits: (I) Detection of a bursty topic as soon as it emerges; (II) Early prediction if the bursty topic is likely to go viral; (III) Summarization of related bursty topics into semantically coherent events that can be monitored; (IV) Contextualization of the events with its temporal evolution and corresponding coverage across other news media.

However, such a system poses a huge challenge to current works on topic model, which can be grouped into three categories. In offline category, noticeably works including *PLSA* and *LDA* [5] model bursts as state transitions. However, they are not suitable to detect new bursty topics which just started to grow viral in real time. Another category [6, 3] is designing some inherent granularity data structure to record the tweet stream, then using data arriving before to infer topics online. However those solutions can't be scaled up to a societal level. Little work regarding real time category, [7] detects events in real-time with predefined keywords which makes it inapplicable to general bursty topic detection. In short, most of existing works focus on high complexity model in offline/online environment.

In this demonstration, we present CLEar(*Cl*airaudient *Ear*)[1] a real-time online platform for detecting, monitoring, summarizing, contextualizing and visualizing bursty and viral events from Twitter. Specifically, CLEar provides the following features:

1. Bursty Topic Detection: Identify bursty topics in real-time as they emerge up to potentially a scale of hundreds of millions tweets per day.

2. Popularity Prediction: Predict trending topic popularity, and remove both noisy and spam bursty topics at an early stage.

3. Event Summarization: Cluster related topics detected within close time periods to form a semantically coherent event. Summarize each event by the most-retweeted tweets of the corresponding topics as well as other key statistics such as influential users and tweet sentiment analysis.

4. Event Contextualization: Provide context for each event by illustrating along a time-line the evolution of the event, and integrate from across different other news media sources related content about the event.

*This work was done when the author was a Research Assistant in Singapore Management University.

[1] http://research.larc.smu.edu.sg/pa/CLEar/

Proceedings of the VLDB Endowment, Vol. 7, No. 13

2. SYSTEM FEATURES

2.1 Bursty Topic Detection

Challenges of real time bursty topic detection arise from the following aspects: (1) How to efficiently maintain proper statistics to trigger detection; (2) How to model bursty topics without examining the entire set of relevant tweets; and (3) How to scale to the massive volume of tweet stream. In our $ICDM'13$ work [8], we proposed a solution called TopicSketch which maintains a data sketch of the accelerations of three quantities at any time stamp t: (1) The whole tweet stream $S''(t)$, (2) Each word $X''(t)$ and (3) Each pair of words $Y''(t)$. The first two provide early signals of popularity surge while $Y''(t)$ are used to infer bursty topics from the keyword correlation embedded. We follow up with an idea similar to MACD(*Moving Average Convergence*) to estimate those quantities.

Besides early detection, this data sketch also contributes to latent bursty topics inference. We model the whole tweet stream as a mixture of multiple latent topic streams and we are interested in identifying the top-K latent topics p_k whose rate $a_k(t)$ is greater than a predefined threshold at any time stamp t. Once a predefined detection criteria is satisfied, we trigger the estimation of the parameters p_k and $a_k(t)$ by solving a constrained optimization problem. The algorithm is detailed in [8].

2.2 Popularity Prediction

The raw result from bursty topic detection are usually laden with noises and spam bursty topics, rendering popularity prediction at an early stage a necessity. The challenges of this problem [4] come from the uncertainty in information diffusion path and insufficient information at the early stage of a burst, offering little clue as to whether the detected bursty topic would sustain its virality or simply die down quickly. On the other hand, user behavior like replying and retweeting provide new mechanism for topic diffusion. We therefore extract a rich variety of features including retweet graph structure features, meta-data features and temporal features. The most influential features are chosen to build a regression model to predict the topic popularity measured by the number of users involved in the topic.

2.3 Event Summarization

In addition to traditional summarization methods which mainly focus on content summarization to extract representative tweets from an event-relevant tweet set [1], we derive three key features as a supplement to facilitate event understanding and visualization. Firstly, due to the existence of many duplicate and semantically close topics from the previous detection step, it is desirable to remove duplicate topics and group together topics corresponding to different stages in the event development. As we detect a bursty topic at a very early stage, for the subsequent short period of time we would monitor the dynamics of its popularity growth to ensure that this topic is indeed drawing an increasing amount of attention before we cluster it with other topics into events. Secondly, frequently-mentioned keywords and Twitter-specific entities are used as surrogates for bursty topics. Thirdly, we analyze and present potential influential users along with sentiment analysis result as auxiliary resources to explore and understand public opinion. Due to the dynamic nature of events, the summarization of each event needs to be updated and refined incrementally by incorporating potential new emergent topics and filtering out old topics that have died off.

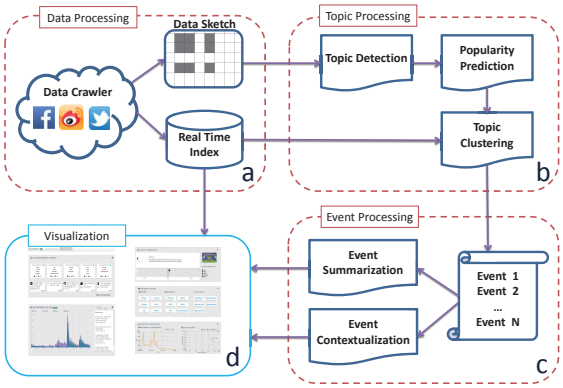

Figure 1: The Architecture of CLEar System

2.4 Event Contextualization

It has been a challenging task to lend event summarization with better readability and user-friendliness. In most cases, users need access to the context to make sense of a topic which is often given as a set of top-ranked keywords. Our event contextualization therefore integrates three components — (1) related news links and (2) characteristic images to provide horizontal context, and (3) event development presented along time-line to provide vertical context. First of all, as rich multimedia content such as images and news links embedded in tweets offers a much broader context to make possible in-depth analysis and deeper understanding of an event, we generate candidate links to news to offer a panoramic view of the development of the same event as covered on conventional media. Secondly, we present representative images of the same event extracted from relevant news media to give a quick glance of the event visually. Lastly, representative tweets are presented chronologically to build up an event time-line which helps users identify the progressive stages of the event.

3. PLATFORM DESIGN

In this section, we describe the platform design of CLEar which consists of a Back-End (dotted line) and a Front-End (solid line) as illustrated in Figure 1.

3.1 Data Processing

Due to Twitter API[2] limit, we select a manageable subset of users which maximizes the information coverage over the Singapore Twitter network and crawl their latest published tweets(Figure 1.a). For each tweet, we preprocess it by lowercasing and removing stop words before passing onto subsequent modules. In parallel, a real-time inverted index is built using open source framework elasticsearch [3] which supports follow-up modules including popularity prediction and event contextualization. A new incoming tweet is available for search as soon as it is added to the inverted index,

[2]https://dev.twitter.com/docs/api
[3]http://www.elasticsearch.org/

which is critically important for time-sensitive tasks such as bursty topic analysis.

3.2 Topic Processing

The data sketch is efficiently updated when a new tweet comes. We use a lazy maintenance technique to reduce the number of accelerations to be updated to $O(H \cdot |d|^2)$, where $|d|$ represents the average number of words in a tweet. Later on, the monitor would compare the current data sketch with historical average, judge whether detection criteria is satisfied and send notification to estimator if needed. Upon notification, the estimator would infer the detected topic (Figure 1.b). All detected bursty topics will be stored in Redis[4] for future reference.

It is worth noting that the number of distinct words can easily reach the scale of millions, resulting in exorbitant data sketch memory consumption in topic inference. We therefore keep a set of active words encountered recently (e.g., last 15 minutes). However, it turns out that the size of this reduced active word set is still too large for real-time processing. Referring to LSH (Locality Sensitive Hashing), we use H hash function to map words into B buckets where B is a number much smaller than N. Consequently, the size of the sketch is reduced to $O(H \cdot B^2)$ and the inference optimization problem is reduced to $O(H \cdot B \cdot K)$, which is computationally feasible in real-time setting. Finally, we use count-min algorithm [2] to recover the probabilities of words from the distribution of buckets.

3.3 Event Processing

Once a bursty topic is detected, the prediction module will first estimate the start time of this topic by tracing back to the very first related tweet. It would then extract those features we proposed using the relevant information between the start time and the detection time. Next, a pre-trained regression model would be used to predict the number of users who would be involved in this topic after x time intervals. Using pre-defined rules such as growth rate, we can learn this topic's trend and filter out meaningless topics. At last, we start a new tracking thread to either cluster this topic into existing event or assign a new event if no semantically related event is found, and then track the evolution of this event dynamically over time. A thread pool is used to execute the event tracking task in parallel.

Finally, a two-level summarization framework (Figure 1.c) is adopted to capture the evolution skeleton of the event along time-line. At content level, we use simple yet effective statistical methods to obtain some key statistics such as the influential users who play a significant role in driving the event forward and the overall sentiment towards this event. At context level, we use a rule-based method to identify the representative images and real-world news links about this event from the links embedded in tweets. The juxtaposition of the information diffusion of an event in online platform like Twitter and the news report coverage of the very same event in off-line real-world offers an interesting perspective to examine the interplay between the two worlds.

3.4 Event Visualization

The "Latest Bursty Topics" panel displays five most recently detected topics (Figure 2.a), each summarized by a set of bursting keywords and given in a cubicle. The arrow

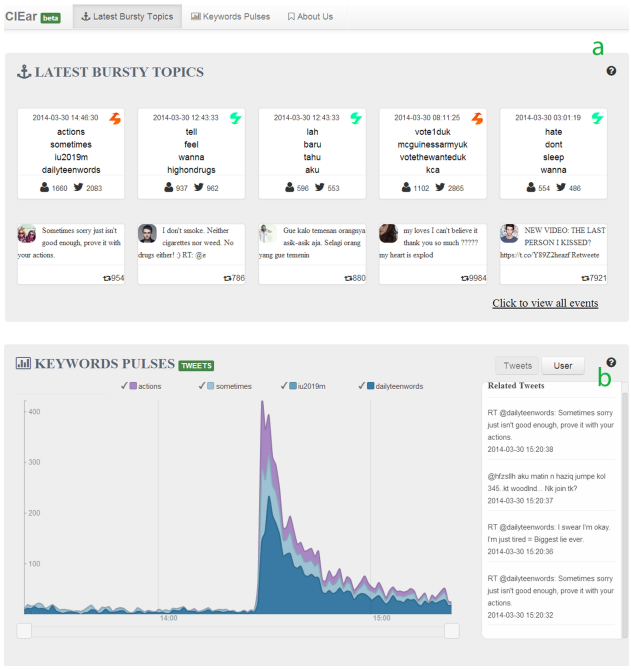

Figure 2: The Latest Bursty Topics and Keywords Pulses of CLEar System

on the top right corner of the cubicle is to indicate the trend of the topic — A red arrow indicates a growing popularity while a green one a waning popularity. The link provided at the right bottom leads to the page where all detected events are displayed (Figure 3.c). Topics associated with the same event are clustered together and listed below each corresponding event.

Based on Rickshaw [5], a real-time monitoring panel (Figure 2.b) is provided to monitor the popularity trajectory of each keyword of the most recent event. The graph is to visualize the number of times every single word associated with the latest detected topic is mentioned in an one-minute interval. On the right, the new tweets of the corresponding topic are updated in real time as they are being generated in Twitter.

Each cubicle on the "Latest Bursty Topics" panel links to the corresponding event details page. It would expand the chosen event into event time-line, word cloud, tweet statistics, sentiment analysis and snapshot. Besides event time-line, an image and a few links to news about this event are provided (Figure 3.a). The word cloud (Figure 3.b) consists of 3 parts, word, hashtags and top frequently-mentioned users. The historical popularity trajectory chart demonstrates the number of tweets and users participating in this event since the right beginning. The influencer chart shows influential users of different sentiment. The larger the bubble, the more influential the user is. Upon clicking on each slice of the pie chart for sentiment analysis, tweets labeled with the specified sentiment would be displayed on the table beside. The snapshot would retrieve the contents in the most frequently-mentioned links and present images and summaries of the content.

[4]http://redis.io/

[5]http://code.shutterstock.com/rickshaw/

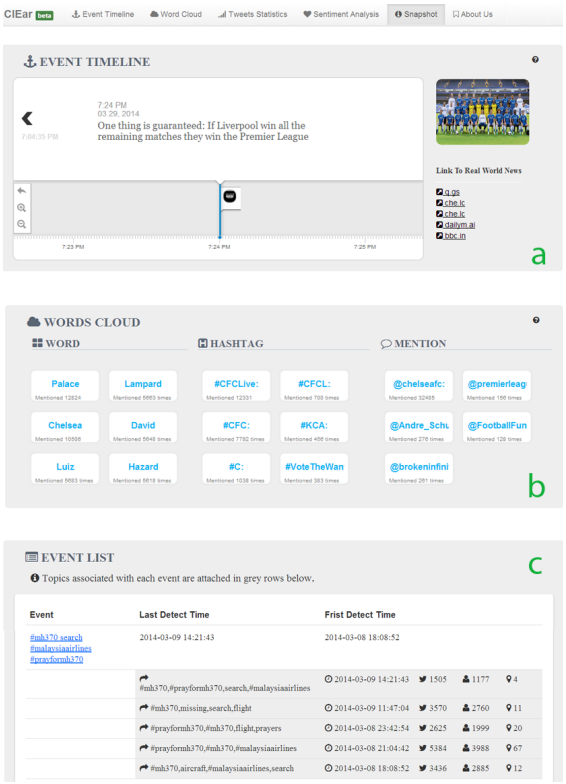

Figure 3: The Event Summarization and Event List of CLEar System

4. DEMONSTRATION

We demonstrate the efficiency and usability of our system in detecting bursty events at early stage and summarizing events on a large scale. All live demonstrations will be conducted on real Singapore Twitter stream. The demonstration is mainly to show the following features of CLEar:

1. A concise yet telling overview of bursty topics detected by TopicSketch, which demonstrates its capacity for real-time processing.

2. The quality of detected bursty topics by popularity prediction module to filter out meaningless topics and the clustering of related topics into coherent events.

3. The dashboard event summarization from both content and contextual perspectives.

We show how CLEar consolidates topics detected into an event. For example, from Figure 3.c, we see that a few topics belonging to event *"MH370 Lose Contact"* were detected between *2014-03-08* and *2014-03-09*. One can tell from the Figure 2.a, the latest topic was detected at 14:46 pm. In the chart below Figure 2.b, we can see a very obvious burst, which means CLEar detected this topic right after it went viral on Twitter. Furthermore, we will show how CLEar can offers a multi-dimensional event summary. On one hand, it would identify influential users and popular tweets in an event, organize these information and present it logically and coherently. On the other hand, with real-time Twitter monitoring, it would give a clear vision and overall picture of how this event bursted, brewed, developed, and subsided. The system would constantly track the detected events so that the vicissitudes of an event would be recorded and presented in an intuitive and accessible manner. Finally, we will share some experiences and lessons learnt from the process of building such a system and some potential improvements.

5. CONCLUSIONS

In this demonstration, we present a real-time system for live event observation on social media, which appeals to users in both academia and industry. To our best knowledge, this is the first unified system to provide bursty topic detection, popularity prediction, event summarization, event contextualization and event visualization in real-time setting.

6. ACKNOWLEDGMENTS

This research has been supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA) and the Pinnacle Lab at Singapore Management University. Chen Lin and Runquan are partially supported by China Natural Science Foundation under Grant No. NSFC61102136, CCF-Tencent Open Research Fund under Grant No. CCF-Tencent20130101, Base Research Project of Shenzhen Bureau of Science,Technology, and Information under Grand No. JCYJ20120618155655087.

7. REFERENCES

[1] D. Chakrabarti and K. Punera. Event summarization using tweets. In *Proceedings of the 5th International Conference on Weblogs and Social Media*, pages 66–73, 2011.

[2] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[3] D. He and D. S. Parker. Topic dynamics: an alternative model of bursts in streams of topics. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 443–452. ACM, 2010.

[4] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *Proceedings of the 20th international conference companion on World wide web*, pages 57–58. ACM, 2011.

[5] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[6] C. Li, A. Sun, and A. Datta. Twevent: Segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 155–164. ACM, 2012.

[7] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[8] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang. Topicsketch: Real-time bursty topic detection from twitter. In *Proceedings of the 13th International Conference on Data Mining*, pages 837–846. IEEE, 2013.