International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) ISSN: 2395-5325
Volume 3, Special Issue 1 (February '2017)
Special Issue on 1st National Conference on Advanced Computing and Communication Technologies,
Conference held at Siri PSG Arts and Science College for Women, Sankari,Tamilnadu,India.

BURSTY TOPIC DETECTION FROM TWITTER USING SPATIO-TEMPORAL HASHTAG CLUSTERING

S.Gajapriya,
Research Scholar,
Department Of Computer Science,
Theivanai Ammal College for Women,
Villupuram,Tamilnadu,India.

K.Manohari,
Assistant Professor,
Department Of Computer Science,
Theivanai Ammal College for Women,
Villupuram,Tamilnadu,India.

Abstract: Twitter Monitor, a system that performs trend detection over the Twitter stream. The system identifies emerging topics (i.e. 'trends') on Twitter in real time and provides meaningful analytics that synthesize an accurate description of each topic. In this paper, we focus on hierarchical spatio-temporal hashtag clustering techniques. Our system has the following features: (1) Exploring events (hashtag clusters) with different space granularity. Users can zoom in and out on maps to find out what is happening in a particular area. (2) Exploring events with different time granularity. Users can choose to see what is happening today or in the past week. (3) Efficient single-pass algorithm for event identification, which provides human-readable hashtag clusters. (4) Efficient event ranking which aims to find burst events and localized events given a particular region and time frame. To support aggregation with different space and time granularity, we propose a data structure called S TREAMCUBE, which is an extension of the data cube structure from the database community with spatial and temporal hierarchy. To achieve high scalability, we propose a divide-and-conquer method to construct the STREAMCUBE. To support flexible event ranking with different weights, we proposed a top-k based index.

Keyword: Topic Sketch, tweet stream, bursty topic, real time

I.INTRODUCTION

In recent years, rates of social media activity have reached unprecedented levels. Hundreds of millions of users now participate in online social networks and forums, subscribe to micro blogging services or maintain web diaries (blogs). Twitter1, in particular, is currently the major micro blogging service, with more than 11 million active subscribers. Twitter users generate short text messages — the so-called 'tweets'—to report their current thoughts and actions, comment on breaking news and engage in discussions. The product of social activity on Twitter reaches an estimated total of over 6M tweets per day. Every tweet is associated with an explicit timestamp that declares the exact time it was generated. Moreover, every user has a well-defined profile with personal information (name, location, biographical sketch).

Such a document stream contains a great wealth of information and offers significant opportunities for exploration, as well as challenges. One of the first challenges that comes to mind, and which we try to address with our system, is to automatically detect and analyze the emerging topics (i.e. the 'trends') that appear in the stream and to do so in real time. Trends are typically driven by emerging events, breaking news and general topics that attract the attention of a 1http://www.twitter.com/ Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. The large fraction of Twitter users. Trend detection is thus of high value to news reporters and analysts, as they might point to fast-evolving news stories. For example, at the announcement of Michael Jackson's death on June 25, 2009, Twitter was immediately flooded with an enormous volume of related commentary. Trend detection is also important or online marketing professionals and opinion tracking companies, as trends point to topics that capture the public's attention. The requirement for real-time trend detection is only natural for a live stream where topics of discussion shift dynamically with time. Furthermore, for such a system to be scalable over massive document streams, an approach is required that makes as few passes over the data as possible. With 500 million tweets (messages) posted everyday, Twitter has become one of the leading social media services around the world. In Twitter, users can post short tweets, which are limited to 140 characters, to share 'what is happening' with their followers. Meanwhile, interesting tweets can be retweeted (re-posted) to propagate further in the social network The tweet stream can be considered as up-to-date news sources of the physical world. For example, during the United States presidential election of 2012, the Twitter Political Index was used to measure users' sentiments towards the candidates. By treating Tweets users as human sensors, [1] can detect earthquakes in real-time by monitoring earthquake related tweets. Furthermore, it is shown that the up and down changes.

In this paper, we propose a novel framework for hierarchical spatio-temporal hashtag clustering over the Twitter stream, which aims to help users explore the Twitter data interactively. An event is considered as a hashtag cluster. For example, the United States presidential election of 2012

can be represented by hashtag cluster {'#Election2012', '#Obama', '#Romney'}, where '#Obama' and '#Romney' represent two main candidates in the campaign respectively. As shown in Figure 1, we organize hashtag clusters into data cubes according to their timestamps and geographical information. Different from traditional data cubes from database literature [3], our framework constructs data cubes from tweet stream in real-time in stock market are correlated with users' moods in Twitter.

Trend Detection and Analysis: Twitter Monitor performs trend detection in two steps and analyzes trends in a third step. First, it identifies 'bursty' keywords, i.e. keywords that suddenly appear in tweets at an unusually high rate. Subsequently, it groups bursty keywords into trends based on their co-occurrences. In other words, a trend is identified as a set of bursty keywords that occur frequently together in tweets. After a trend is identified, Twitter Monitor extracts additional information from the tweets that belong to the trend, aiming to discover interesting aspects of it. Each of the three steps described above is pictured as a component of the diagram shown in figure 1 and is described in detail in the following paragraphs.

Detecting Bursty Keywords: A keyword is identified as bursty when it is encountered at an unusually high rate in the stream. For example, the keyword 'NBA' may usually appear in 5 tweets per minute, m yet suddenly exhibit a rate of 100 tweets/min. Such 'bursts' in keyword frequency are typically associated with sudden popular interest in a particular topic and are often driven by emerging news or events. For example, a sudden rise in the frequency of keyword 'NBA' may be linked to an important NBA match taking place. Twitter Monitor treats bursty keywords as 'entry points' for trend detection. In other words, whenever a keyword exhibits bursty behavior, Twitter Monitor considers this an indication that a new topic has emerged and seeks to explore it further (more on that in Sections 2.2, 2.3). Effective and efficient detection of bursty keywords is thus crucial to Twitter Monitor's performance. To detect bursty keywords, we developed a new algorithm, Queue Burst, with the following characteristics: (I) One-pass. Stream data need only be read once to declare when a keyword is bursty. (II) Real-time. Identification of bursty keywords is performed as new data arrives. No optimization over older data is involved. (III) Adjustable against 'spurious' bursts. In some cases, a keyword may appear in many tweets over a\ short period of time simply by coincidence. The algorithm is tuned to avoid reporting such instances as real bursts. (IV) Adjustable against spam. Spam user groups repetitively generate large numbers of similar tweets. The algorithm is tuned to ignore such behavior. (v) Theoretically sound. Queue Burst is based on queuing theory results.

Trend Analysis: After a trend is identified as a subset Kit of bursty keywords, Twitter Monitor attempts to compose a more accurate description of it. The first step towards this end is to and submit their own short description of a trend (Figure 2). Specifically, trends can be ranked by volume, recency or a combined score of the two. Moreover, user-created descriptions are stored on Twitter Monitor and displayed on the website as long as there are enough of them that exhibit significant overlap. Reported trends are also accompanied by a small sample of representative tweets and a link to Twitter's live stream for the trend Finally, Twitter Monitor uses an additional tab to display daily trends, i.e. trends that have emerged within the last day, ranked by aggregate volume of tweets. Clustering is one of the widely-used tools for news aggregation. However, it is deficient in three regards: the number of clusters is a linear function of the number of days (assuming that the expected number of storylines per day is constant), yet models such as Dirichlet Process Mixtures (Antoniak 1974) only allow for a logarithmic or sub linear growth in clusters. Secondly, clusters have a strong aspect of temporal coherence. While both aspects can be addressed by the Recurrent Chinese Restaurant Process (Ahmed and Xing 2008), clustering falls short of a third requirement: the model accuracy does not improve in a meaningful way as we obtain more data — doubling the time span covered by the documents simply doubles the number of clusters. But it contributes nothing to our understanding of longer-term patterns in the documents.

II. RELATED WORK

Our problem is related to work done in the topic detection and tracking community (TDT), which focuses on clustering documents into stories, mostly by way of surface level et al. 2004). Moreover, there is little work on obtaining two-level organizations (e.g. Figure 3) in an unsupervised and data-driven fashion, nor in summarizing each story using general topics in addition to specific Data Cube. Our work is mainly inspired by the data cube [3] from the database literature, which is used for organizing and exploring multidimensional data with operations like slice, dice, drill up/down. Our work can be considered as an extension of the traditional data cube for Twitter data.

There are several differences: (1) STREAM CUBE is constructed incrementally over the Twitter stream according to a time hierarchy and a space hierarchy. (2) The semantic meaning of each cube is different. We aim to provide human-readable events (i.e., clusters of hash tags). (3) Given a cube with respect to a particular time frame and region, we aim to provide a meaningful ranking (finding burst events and local events), which is not considered in the traditional data cube. Clustering Techniques for Social Media. Existing work on clustering techniques for social media can be classified into three categories: (1) Tweet-based clustering [4]–[7], which is an extension of the traditional text clustering. Different from high quality news, tweets are short and noisy. To address the problem, they either argument the tweets with external knowledge base [4], term relatedness [5], or propose a more advanced similarity measurement [7]. (2) Burst keyword- based clustering [1,8]. These work focus on the burst keywords instead of single tweets. However, this will only cover partial events with high burstiness. (3) Hashtag-based clustering.

The common drawback of tweet-based clustering and burst-keyword-based clustering is that their results are less human readable. Our work defines an event as a cluster of hashtags, which are more expressive since they are user provided keywords. Moreover, our hashtag-based clustering can

overcome the drawbacks of noisy tweets in a more elegant way. To the best of our knowledge, hash tag clustering has not been well studied. The most relevant work are [9] and [10]. Event Ranking. We are particularly interested in finding burst events and local events. [11] Proposes a sketch-based approach for detecting localized events. [12] also employ the sketch structure for burst detection. Given a term, [13, 14] identify its bursting time interval and region in polynomial time. The common drawback of these works is that they aim to find burst patterns or localized patters from the whole dataset, where only a fraction of events can be detected. In contrast, our work aims to give an overall ranking for all the events in the chosen cube w.r.t. to a particular time frame and region. Recent work on topic models has focused on improving scalability; we focus on sampling-based methods, which are most relevant to our approach. Our approach is most influenced by the particle filter of Canini et al. (2009), but we differ in that the high-order dependencies of our model require special handling, as well as an adaptation of the sparse sampler of Yao et al. (2009).

III. PROPOSED METHODOLOGY

In this paper, we propose a novel framework for hierarchical spatio-temporal hashtag clustering over the Twitter stream, which aims to help users explore the Twitter data interactively. An event is considered as a hashtag cluster. For example, the United States presidential election of 2012 can be represented by hashtag cluster represent two main candidates in the campaign respectively. As shown in Figure 1, we organize hashtag clusters into data cubes according to their timestamps and geographical information. Different from traditional data cubes from database literature [3], our framework constructs data cubes from tweet stream in real-time. Thus our framework is called STREAMCUBE. STREAMCUBE has the following features: Exploring events in different time granularity. Depending on the data analysis purpose, different users may have different requirements on the time granularity. If we want to summarize the top-10 events of a year, we may choose year as the basic time granularity.

In this case, the recent B of an event is not very important. However, if we want to detect emerging events from the stream, we may choose a finer granularity. Exploring events in different space granularity. If we focus on the world-wide breaking events, we may choose global space as our basic space granularity. If we focus on localized news, we may choose district or city as our space granularity. Detecting burst events and localized events. Given a time frame (e.g., today), users may want to know what events break out during that time. Given a region (e.g., the current region shown on the Google map), users may want to know the local events with respect to this particular region. Finally, given a time frame and a region, we can also find the burst localized events. Effective and efficient detection of bursty keywords is thus crucial to Twitter Monitor's performance. To detect bursty keywords, we developed a new algorithm, Queue Burst, with the following characteristics: (I) One-pass. Stream data need only be read once to declare when a keyword is bursty. (II) Real-time. Identification of bursty keywords is performed as new data arrives. No optimization over older data is involved. (III) Adjustable against 'spurious' bursts. In some cases, a keyword may appear in many tweets over a short period of time simply by coincidence. The algorithm is tuned to avoid reporting such instances as real bursts. (IV) Adjustable against spam. Spam user groups repetitively generate large numbers of similar tweets. The algorithm is tuned to ignore such behavior. (V) Theoretically sound. Queue Burst is based on queuing theory result.

IV. SYSTEM MODEL

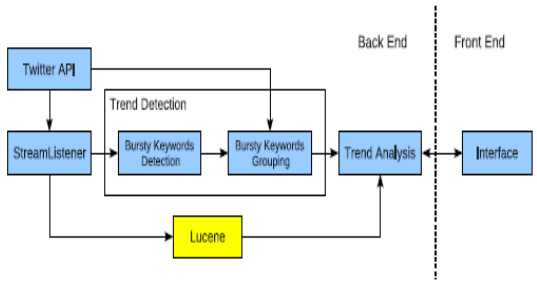

Figure 1: TwitterMonitor Architecture

V. WORKING PRINCIPLE

A) Stream Cube: The top-5 events detected by STREAMCUBE are: (1) 'Grammys', which are famous music awards for recognizing outstanding musicians. The ceremony is broadcast on January 26. (2) 'Peoples Choices', which is an award for honoring the best in popular culture and is presented on January 8. (3) 'HappyNewYear', which is a hashtag used for celebrating the new year of 2014. (4) 'EXABelievers', and 'EXADirectioners', which are two hashtags used for voting musician Justin Bieber, and music band One Direction on a show on a radio station called EXA during the second week of January. (5) 'GoldenGlobe', which is an award in film and television industry and is presented on January 12. The bold hashtag represents the one with the biggest size in the cluster. As we can see, all these events are quite human readable and unique to the time frame of January. Also we can see that hashtag clusters are self-explainable in summarizing events. For example, from the cluster'Grammys2014', 'Lorde', 'DaftPunk'} we can infer that the Lorde (a singer) and Daft Punk (musical group) have performed in the Grammys ceremony and have attracted much attention from the users in Twitter. Another example is {'Golden Globes', 'Breaking Bad', 'American Hustle'}, where we can infer that Breaking bad (a TV show) and American Hustle (a movie) have won the Golden Globes awards.

B) SMCA : As the top-5 events detected by SMCA are: (1) 'Peoples Choices'. (2) 'TeamFollowBack', which is a long last popular hashtags used for gain followers. (3) 'game insight', which is a hashtag for discussing iPod and android games. (4) 'Now playing', which is used for denoting what the user is currently listening to, watching or playing. (5) 'Grammys'. The main problem of SMCA is that popular

events may not carry valuable information. For example, 'now playing' is popular in every month while 'Golden Globes' detected by STREAMCUBE only happens in January. When we ranking the events in January, it is more reasonable to rank 'Golden Globes' higher than 'now playing'. The same problem exists for 'TeamFollowBack', which happens in every month and carries less information. The main reason for such ranking is that SMCA is designed to cluster hashtags instead of find burst events and local events. Thus the final ranking is mostly dominated by popular events.

C) TWITTER MONITOR: TWITTERMONITOR is designed to detect hot events base on burst keywords. The top-5 events detected are: (1) 'ExaBeliebers', which is a hashtag to support Justin Bieber (a singer) for a ratio session. (2) 'Belletstalk', which represents a fund for supporting mental health organizations. (3) 'This Could Be Us utYouPlayin', which is used to highlight awkward photographs of couples, (4) 'JamesFollow', which is used to support James and is heavily used his fans, and (5) 'Grammys'. TWITTERMONITOR mainly has two drawbacks:

(1) Burst keywords have lower quality and carries less information compared with burst hashtags. For example, TWITTERMONITOR clusters 'Belllettalk' with 'mental'. In contrast, STREAMCUBE clusters 'Belllettalk' with hashtag 'mental health', which make it easier for users to understand that 'Bellelettalk' is an event related to 'mental health'.

(2) Considering burstiness is not enough for a good event ranking. For example, TwitterMoniter ranked 'JamesFollow' among the top-5 events in January, which is a hashtag for encouraging people to follow a user called James.

VI.SUMBLR

All the top-5 hashtags have been introduced before. The first problem of SUMBLR is that the events does not have clear boundaries and it is easy to mix different events. Since tweets are extremely short, the clustering is easy to be affected by noisy tweets. For example, 'People's Choice' and 'GlodenGlobes' are clustered into one cluster, but they are two independent events. The second problem is words are not easy to understand. For example, 'Grammys' are clustered with 'perform' and 'love'. However, the 'love' here refers to a song performed in Grammys called 'the same love'. Finally, SUMBLR has problem in finding valuable events that are particular in January, which is similar to SMCA

Algorithm 1: Spatial-temporal Aggregation

Input: $D=\{d_1, d_2, ..., d_n\}$: tweet stream from the current time frame t
Output: The updated STREAMCUBE

```
for each tweet d ∈ D do
    cube ← cubes[t][d.region]
    cube.hashtag-clustering(d)
    cube.update-event-ranking()
for each space level sl from bottom to top do
    for each region at space level sl do
        cubes[t][region] ← spatial-merge(cube,
        cube.children())
for each space level sl from bottom to top do
    for each region at space level sl do
        cubes[2t][region] ←
        temporal-merge(cubes[t − 1][region],
        cubes[t][region])
```

VII. EXPERIMENTAL RESULT

In our demonstration, we will display the online version of Twitter Monitor, with all features described in this proposal and give the audience the chance to perform in-depth inspection of recent Twitter trends. Every trend will be represented by the entities involved or, in absence of identified entities, by the related bursty keywords. The audience will have the option to use the interface in order to acquire more information about trends they deem interesting. In particular, they will be shown additional keywords that are correlated with a trend and skim through representative tweets in order to obtain a better understanding of the related discussion. Moreover, they will be able to track a trend's popularity over time and spot the origin of geographically focused trends. Finally, the audience will interact with the system by ranking the displayed trends according to different criteria and submitting their own descriptions to the system Similar functionality will be provided for daily trends. In parallel, we will describe Twitter Monitor's functionality and architecture, as well as the algorithms that are employed in different components of the system. In general, we will share our experience from building a dynamic monitoring system, the design choices we made and the challenges we faced during development ('spurious' bursts, spam, etc). In case of low connectivity, a back-up scenario will be in place. The back-up scenario will include a simulated run of the system over older, locally stored data. Due to the huge volume of tweet stream, existing topic models can hardly scale to data of such sizes for real-time topic modeling tasks. We developed a "sketch of topic", which provides a "snapshot" of the current tweet stream and can be updated efficiently. Once burst detection is triggered, bursty topics can be inferred from the sketch efficiently. Compared with existing event detection system, from a different perspective – the "accelerations of topics", our solution can detect bursty topics in real-time, and present them in finer-granularity

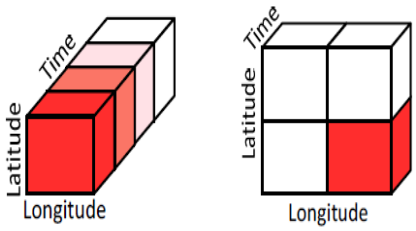

Fig 2. Localness

STREAMCUBE can be extended in many directions. Firstly, we can extend STREAMCUBE to support topic-based exploration. For example, users explore events in a specific domain such as politics, music, travel, breaking news, etc. This can be considered as a four-dimension data cube (i.e., time, latitude, longitude, and topic). Secondly, it is also important to develop an alert mechanism to push information to users from the server instead of waiting for user-initiated queries since users cannot keep monitoring what is happening.

VIII.CONCLUSION

In this paper, we proposed STREAMCUBE to support hierarchical spatio-temporal hashtag clustering, in which case users can explore twitter data interactively with different time and space granularity. To the best of our knowledge, this is the first framework to support such application. Our approach contains three components: (1) A spatio-temporal hierarchy inspired by the quad-tree and the data cube. Hashtag clustering is performed according to a divide-and-conquer strategy at the lowest level of the hierarchy. Then clustering results are merged incrementally in a bottom-up manner. Out-dated clusters are flushed to disk to save space. (2) A single-pass hashtag clustering algorithm. Different from existing clustering techniques, we are dealing with content-evolving hashtags. (3) Event ranking, which is designed to help users identify local events and burst events.

IX.REFERENCES

[1] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in WWW, 2010.

[2] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," Journal of Computational Science, 2011.

[3] J. Han, Data Mining: Concepts and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[4] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen, "Short text conceptualization using a probabilistic knowledgebase," in Proc. of IJCAI, 2011.

[5] O. Ozdikis, P. Senkul, and H. Oguztuzun, "Semantic expansion of tweet contents for enhanced event detection in twitter," ASONAM 2012, vol. 0.

[6] L. Shou, Z. Wang, K. Chen, and G. Chen, "Sumblr: continuous summarization of evolving tweet streams," in SIGIR, 2013, pp. 533–542.

[7] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Realworld event identification on twitter," in ICWSM, 2011.

[8] M. Mathioudakis and N. Koudas, "Twitter monitor: trend detection over the twitter stream," in SIGMOD Conference, 2010, pp. 1155–1158.

[9] C. I. Muntean, G. A. Morar, and D. Moldovan, "Exploring the meaning behind twitter hashtags through clustering," in BIS (Workshops), 2012.

[10] S. Carter, M. Tsagkias, and W. Weerkamp, "Twitter hashtags: Joint translation and clustering," in Web Science 2011. ACM, 2011.