StreamExplorer: A Multi-Stage System for Visually Exploring Events in Social Streams

Yingcai Wu, Zhutian Chen, Guodao Sun, Xiao Xie, Nan Cao, Shixia Liu, Weiwei Cui

Abstract—Analyzing social streams is important for many applications, such as crisis management. However, the considerable diversity, increasing volume, and high dynamics of social streams of large events continue to be significant challenges that must be overcome to ensure effective exploration. We propose a novel framework by which to handle complex social streams on a budget PC. This framework features two components: 1) an online method to detect important time periods (i.e., subevents), and 2) a tailored GPU-assisted Self-Organizing Map (SOM) method, which clusters the tweets of subevents stably and efficiently. Based on the framework, we present StreamExplorer to facilitate the visual analysis, tracking, and comparison of a social stream at three levels. At a macroscopic level, StreamExplorer uses a new glyph-based timeline visualization, which presents a quick multi-faceted overview of the ebb and flow of a social stream. At a mesoscopic level, a map visualization is employed to visually summarize the social stream from either a topical or geographical aspect. At a microscopic level, users can employ interactive lenses to visually examine and explore the social stream from different perspectives. Two case studies and a task-based evaluation are used to demonstrate the effectiveness and usefulness of StreamExplorer.

Index Terms—Social media visualization, visual analytics, social stream, streaming data, self-organizing map.

1 Introduction

Social streams cover an extensive spectrum of ongoing topics on events happening around the world [29], [47]. Hence, the timely analysis and tracking of social streams has become increasingly important to various applications. By monitoring their social streams, decision makers can maintain a high level of situational awareness and appropriately react to major crises in a timely manner [35]. The abundant user-generated information also brings new opportunities for sociologists to conduct data-driven research [3]. Therefore, effective approaches are required to fully support the analysis and monitoring tasks of social streams.

Various visualization systems, such as Whisper [10] and Visual Backchannel [17], have been developed to visualize social streams. However, most systems are generally not scalable for tracking and exploring large events with many live topics. Various event detection techniques have been used to alleviate the problem by extracting more important topics in certain spatio-temporal ranges. Nevertheless, methods based on topic modeling [12] or clustering [43] are usually too computationally intensive to handle live streams on budget PCs. Other methods based on term tracking [37] can deal with streaming data efficiently but do not support an in-depth, topic-based analysis. Hence, the process of tracking and exploring large events with many live topics from social streams on budget PCs in a timely, manageable, and comprehensible manner remains a challenging task.

- *Yingcai Wu and Xiao Xie are with State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. E-mail: {ycwu,xxie}@zju.edu.cn.*
- *Zhutian Chen is with Hong Kong University of Science and Technology, Hong Kong. E-mail: zhutian.chen@outlook.com.*
- *Guodao Sun is with Zhejiang University of Technology, Hangzhou, China. E-mail: godoor.sun@gmail.com.*
- *Nan Cao is with the College of Design and Innovation, Tongji University, Shanghai, China. E-mail: nan.cao@gmail.com.*
- *Shixia Liu is with the School of Software, Tsinghua University, Beijing, China. E-mail: shixia@tsinghua.edu.cn.*
- *Weiwei Cui is with Microsoft Research, Beijing, China. E-mail: weiwei.cui@microsoft.com.*

To overcome the difficulty, the present study aims to make three contributions as follows. Our first contribution is a new framework that processes social streams efficiently to support interactive visualization. The framework can be deployed on a commodity computer and features two elements: (1) a rapid online algorithm that continuously detects important time periods (i.e., *subevents*), and (2) a GPU-assisted *Self-Organizing Map* (SOM) method that can be invoked on demand to efficiently extract topics of tweets made on any subevent.

Our second contribution is the provision of a multi-level visualization method that integrates a novel glyph-based timeline visualization, a map visualization, and interactive lenses to enable an intuitive, multi-faceted analysis of social streams. The timeline visualization visually summarizes important subevents at a macroscopic level, using a combination of glyph-based trend and tree visualizations. It does not only reveal the dynamic changes of a social stream in the context of its past evolution, but also organizes past subevents in a hierarchical manner for easy review and navigation of subevents. For further analysis at a mesoscopic level, the map visualization shows a topic or geographic map for any subevent selected from the timeline visualization. Interactive lenses, such as word lens and network lens, allow users to visually examine the map visualization for an in-depth, multi-faceted analysis at a microscopic level.

The third contribution is a new multi-stage system that is based on the proposed framework and visualization techniques. The system enables end users to track, explore, and gain insights into social streams at different levels. The efficient framework and multi-level visualization make the system scalable to the large and fast social streams as well as manageable for end-users to use on budget PCs.

2 Related Works

This section reviews a few research areas that are closely related to our work of visual analysis of social streams.

2.1 Streaming Data Visualization

Temporal data visualization has been extensively studied [9], [18] and applied in various applications, such as sport analysis [31] and social media analysis [34]. Streaming data visualization is an important research area in temporal data visualization - one that typically deals with continuously updating, unbounded data sequences [25]. Different methods have been introduced to visualize streaming text data [5], [22]. CloudLines [28], a novel compact visualization of event episodes, has been presented to visualize dynamic time series for the exploration and monitoring of streaming news data. Researchers also presented a comprehensive taxonomy of dynamic data visualization to help users understand the relationship between the changes in data and the interpretability of visual representations [14]. Several visual metaphors, such as storylines [42] and sedimentation [25], [33], have also been proposed to visualize live data streams.

However, the existing techniques that are mainly used for streaming text data may not work effectively for social stream data with short text and rich multi-faceted information. In addition, a live social stream is usually changing at a significantly rapid rate. Existing methods that continuously update their views require significant cognitive effort to monitor the rapid changes. Furthermore, the lack of an intuitive mechanism to archive important changes for further exploration is considered an obstacle in the application of techniques that can track fast-changing social steams.

2.2 Social Media Visualization

Social media visualization has attracted considerable attention in recent years [16], [24], [44], [48]. Many proposed methods usually extract places [35] and time [16] from Twitter data, and then visually summarize such data by aggregating the messages into places and time. Existing systems also provide topical overviews of Twitter data [32], [39], [45]. However, these methods lack sufficient visualization and analytics support to extract and visualize the ongoing topics in live social streams. For instance, Steiger et al. [39] has introduced a geographic, hierarchical SOM (Geo-H-SOM) to extract spatiotemporal and semantic clusters of Twitter data to provide a topical overview in the spatio-temporal context. However, Geo-H-SOM cannot handle live social streams for three reasons. First, it produces a sequence of SOMs in different timestamps with topic clusters that are randomly distributed, making the visual monitoring and tracking tasks difficult. Second, the computation of semantic, geospatial, and temporal similarity among tweets in Geo-H-SOM is time consuming, along with the requirement of SOM algorithm computation. Third, the semantic similarity, which is computed based on Latent Dirichlet Allocation (LDA) [7], requests a predefined number of topics.

At the same time, visual analytics of information diffusion on social media has been the subject of increasing attention. Researchers have employed novel visual metaphors, such as sunflower [10], ripple [44], and river [41], to visualize information diffusion on social media. Among the methods, only Whisper [10] supports visualization of real-time diffusion in live social streams. However, Whisper does not support topic-based visualization and represents every tweet as a seed in a sunflower. This limitation makes it difficult to scale it up because drawing all the tweets in the sunflower can lead to serious visual clutter.

Thus far, only a few systems have been shown to handle live social streams. Dörk et al. [17] introduced Visual Backchannel, which utilizes a tailored stacked graph to visualize the topical changes of an event over time. ScatterBlogs2 [8] enables users to interactively create task-specific filters to retrieve highly relevant tweets from social streams for further analysis. TwitterScope [20] groups the messages of a social stream into clusters and displays the clusters in a dynamic map. It models a social stream as a dynamic graph, with its nodes and edges encoding the messages and their similarities, respectively. A dynamic graph layout algorithm and Procrustes projection are used to ensure visual stability of the map layout. Our method also produces a dynamic topic map of a social stream, but with the GPU-based SOM algorithm. The above methods require users to constantly follow socials stream without detecting and emphasizing critical moments; thus, users are prone to miss significant patterns and feel that the task is tedious.

2.3 Event Detection in Social Media Visualization

Numerous event detection methods [38], [40] have been proposed. Interested readers can refer to a recent survey [21] for a complete review. This section mainly discusses the methods used in existing visual analytics systems.

Topic modeling, such as LDA and probabilistic models [6], [7], which discovers main themes in document collections has been employed to detect events. For example, Chae et al. [12] employed LDA to extract and rank major topics. A seasonal-trend decomposition procedure based on Loess smoothing (STL) was employed to compute abnormality scores (z-score) for the top-ranked topics. ScatterBlogs2 [8] can cluster tweets into topics using an LDA method, and uses a list of small tag clouds to visually represent the topics. However, previous methods [8], [12] cannot reveal the relationship among topics. Moreover, the methods based on LDA are computationally intensive and the number of topics must be specified by users.

Meanwhile, incremental clustering methods have also been used. Thom et al. [43] introduced an incremental clustering method based on an enhanced Lloyd scheme to detect spatiotemporal clusters of term usage. Liu et al. [33] developed TopicStream, which combines the strengths of an evolutionary tree clustering model, a streaming tree cut algorithm, and a sedimentation metaphor to visually analyze hierarchical topic evolution. Our tailored SOM algorithm is an incremental clustering method, but it is accelerated by GPU to handle live streams more efficiently.

Term tracking methods use keywords that are automatically identified from other channels [4] or those that are manually defined by users [17], [37] to track and detect social events. Twitincident detects incidents from emergency broadcasting services [4]. TwitInfo uses a congestion control mechanism to identify the peaks of high tweet activity [37] . Our proposed approach uses a similar algorithm to automate subevent detection. Visual Backchannel

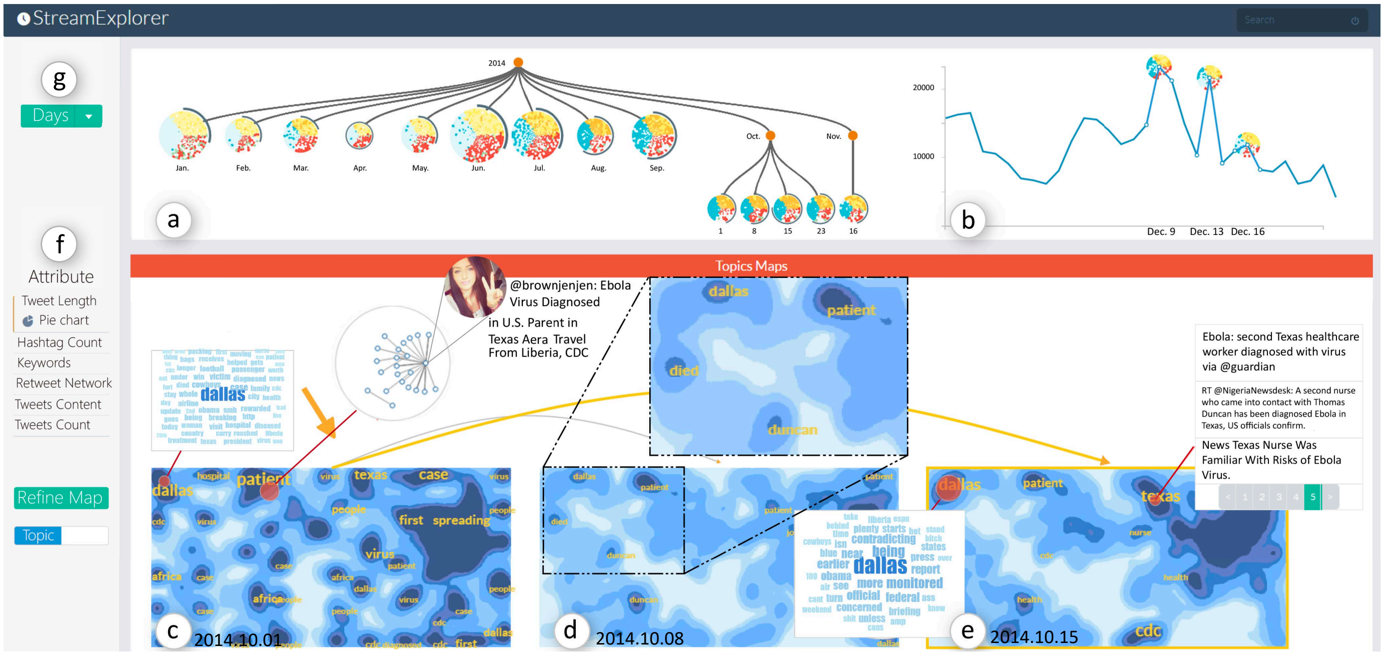


Fig. 1. User interface of StreamExplorer: a timeline visualization with the combination of (a) a visual tree of aging subevents and (b) a line chart of recent subevents; (c)-(e) three topic maps with a set of interactive lenses; (f) a panel for choosing interactive lenses; (g) options of time units.

treats frequent keywords as topics and uses a stacked graph to visualize topic evolution [17]. However, keywords may not adequately characterize the topics. Understanding the topics with many different, highly rated keywords without sufficient context is difficult. Understanding massive tweets using only term tracking methods is difficult, because topic-based analysis is not supported

Our framework integrates a term tracking method to detect subevents and an incremental clustering method to reveal topics in those subevents. The framework maximizes the advantages and bypasses the disadvantages of the two methods. Additionally, it is more efficient and scalable, because it mostly employs the light-weight subevent detection algorithm to cope with live streams. The GPU-assisted SOM algorithm is triggered on demand for important subevents selected by users for further examination. Moreover, it can enable a multi-stage visualization system, with which users do not have to constantly monitor and track the live streams.

3 User Interface

Figure 1 shows our user interface with two major views: a timeline visualization (top part) for displaying the dynamic changes in tweet volume and a map visualization (bottom part) for exploring the social stream from a geographic aspect (Figure 8(c)-(e)) or topical aspect (Figure 1(c)-(e)). A user can define an ***event*** that he wants to follow and analyze by providing one or a few keywords in the search bar located at the top right of the user interface (Figure 1(h)). A user is allowed to add, remove, or modify the specified keywords in the search bar of the interface.

The timeline visualization contains a line chart (Figure 1(b)) and a tree visualization (Figure 1(a)), which provides an immediate overview of what is going on about an event at a macroscopic level. The line chart is used to show tweet activity (i.e., the trend of tweet volume). Recent subevents, namely, critical time periods (called ***subevents***), are highlighted using a DICON glyph [11] to show the multi-faceted visual summary of the tweets in the subevent. The aging subevents will eventually fade out from the left of the line chart and be aggregated into the rightmost node of a subevent tree (Figure 1(a)), such that the sedimentation of the subevents can be intuitively revealed. The tree organizes the past subevents hierarchically to facilitate the exploration and navigation of past subevents.

For further analysis at a mesoscopic level, the map visualization (located at the bottom of Figure 1) displays a topical or geographic summary of the tweets in a subevent selected by a user from the timeline visualization. Regions with a dark color represent the highly concentrated tweets. The user is allowed to compare multiple subevents or track the content/geographic changes of the event in two ways: 1) he can select multiple subevents and create a series of maps accordingly, or 2) he can create a single map (Figure 1(c)) for a specific subevent as filter, and then use the filter to generate maps ((Figure 1(d) and (e) linked to the filter map using arrows) for other subevents.

The user can further drag various interactive lenses, such as word lens and bar lens, from the lens panel (Figure 1(f)), and drop the lenses to any area on the map visualization to inspect the area from various perspectives. The interactive lenses thus enable the in-depth and multi-faceted analysis of a subevent at a microscopic level.

4 A Multi-Stage Framework For Processing Social Streams

This section presents the multi-stage framework and its two components, namely, a subevent detection algorithm and an SOM method.

4.1 Framework

Tracking and understanding the unfolding of an event is difficult because of the highly dynamic, large-scale streaming data. Numerous computational resources are often requested by existing systems [12], [33] to fully process such data. Other systems based on term tracking [17], [37] can efficiently process such data, but lack adequate support for the in-depth, topic-based analysis and visualization. Moreover, constantly updating a visualization of the processed data without a proper strategy would easily lead to information overload of analysts. Thus, we introduce a multi-stage framework to reduce the computational overload of computers and the information overload of analysts.

Figure 2 shows the framework consisting of three parts: a ***subevent detector*** for detecting subevents, a ***preprocessor*** for processing the collected tweets in the detected subevents, and a ***map generator*** for producing topic maps for the subevents selected by a user. The relevant tweets in the detected subevents are processed in the data preprocessor to extract word vectors and stored in a database.

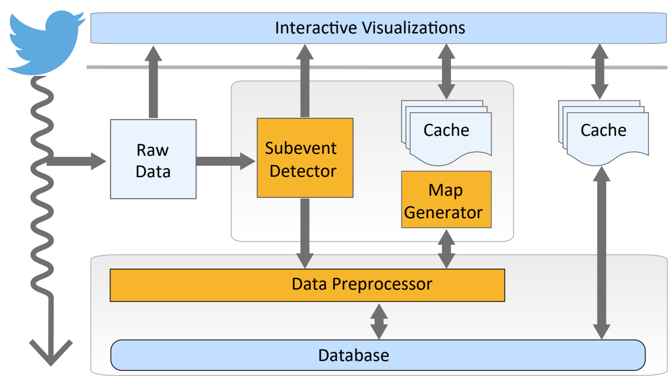


Fig. 2. Framework with four main components: subevent detector, preprocessor, map generator, and interactive visualizations.

The preprocessor and map generator are relatively expensive and significantly affect system performance without a proper strategy. To cope with the problem, we employ a call-by-need strategy, which relies on the subevent detector and interactive visualizations to serve as gate keepers. This strategy also determines the portions of the data to be processed. Therefore, StreamExplorer couples a fast subevent detection method and user interactions with an "expensive" analytical mining, which occurs once a user decides to investigate a given slice of data.

4.2 Subevent Detection

Finding and highlighting important subevents can greatly reduce the efforts exerted in tracking social streams (see **T1** in Section 5.1). We regard the time periods of a social stream with high tweet activity as important subevents. Our method detects the subevents characterized by an unusually high volume of tweets, and then measures the diversity of the identified subevents.

Our system identifies a subevent from streaming Twitter data based on a congestion control mechanism used in TwitInfo [37], which is highly efficient with reasonable precision and recall rates. The method employs exponentially weighted moving average and variance with $\alpha = 0.125$ to determine whether an unusually large number of tweets are arriving. Specifically, a new window starts when a significant increase in tweet count occurs compared with the historical mean. Following TwitInfo, the current work identifies the significant increase when the tweet count is twice as many as the historical mean. Such a ratio can be considered as the sensitivity threshold of the algorithm. The window ends when the tweet count returns to the same level as when it has started, or when a new significant increase in tweet count is detected. Meanwhile, an event peak is defined as the moment when the tweet count reaches the maximum in a given time window. All the tweets generated within the time window are considered to belong to the corresponding subevent. Additional details on the algorithm can be found in [37].

Although the algorithm can detect subevents with high tweet activity, it only handles the streaming tweet as a purely digital signal, without considering its semantic content. To address this problem, we further evaluate the entropy of a subevent using an information-theoretic measure. The higher entropy a subevent exhibits, the more diverse the subevent is. The entropy can be computed as follows:

$$H(X) = -\sum_{x \in X} p(x) \log p(x)$$

where X represents the words in a subevent, and $p(x)$ is the probability of word x in the subevent.

4.3 Self-Organizing Map

Our system employs an SOM algorithm to cluster tweets and provide a topical summary for a subevent. We use clustering instead of classification to find topics, because major events on Twitter can develop rapidly with different emerging or disappearing topics over time. Traditional classification methods, such as support vector machine used in ScatterBlog2 [8], work effectively for relatively stable streams, such as a flood event, with known topics. We use data clustering, because unexpected topics are likely to be ignored by classification methods.

Although numerous clustering algorithms have been proposed, we select SOM for four reasons. First, the algorithm maps the tweets to a 2D map, which naturally accommodates the intuitive visual metaphor of lenses on the map for multi-faceted and in-depth exploration. Second, an SOM preserves the topology of the data (i.e., local neighborhood relations) and does not impose a hard partition on such data, thereby clearly revealing the relative or qualitative mutual relationships among the tweet clusters. Third, the algorithm is parallel in nature and can be easily accelerated, thus enabling the interactive visualization of streaming data on a personal computer equipped with a commodity GPU. Fourth, a well-trained SOM can also be used as a classifier that can track fast topical changes efficiently.

An SOM is defined as a series of neurons organized in a 2D grid. Each neuron has a weight vector with the same dimension as the input vectors. The weight vector is initialized randomly. At iteration t, the algorithm selects input vector x_i, and then finds the neuron with the shortest Euclidean distance to x_i. The neuron is called *Best Match*

Unit (BMU). The weights of the adjacent neurons of the BMU are updated toward x_i. The degree of the update decreases with the iterations and the distance from the BMU. The above procedure repeats until the map becomes stable. The result is a trained SOM.

4.3.1 Accelerating SOM

The system should efficiently produce an SOM for tens of thousands of tweets. However, the computation of the basic SOM algorithm is time-consuming. Therefore, we use the batch-type SOM [27], a variant of the basic SOM, in our system. In the batch-type algorithm, the training set is gone through for one time, which is called an epoch. The weight vector of neuron u at epoch t is updated as follows:

$$w_u(t) = \frac{\sum_{i=i_s}^{i_f} h_{b_i,u}(t) x_i(t)}{\sum_{i=i_s}^{i_f} h_{b_i,u}(t)} \tag{1}$$

, where i_s and i_f denote the start and end indices of the input samples at t, respectively; b_i is the BMU for data vector i; and $h_{b_i,u}$ is the neighborhood function, which is taken as a Gaussian function, thus ensuring that the magnitude of the update decreases with the distance from b_i to neuron u. We note that the width of the neighborhood function (i.e., standard deviation) decreases monotonically with t. The weight vector of each neuron can be computed in parallel at each epoch. Thus, the algorithm is significantly faster than the basic sequential SOM.

When the clustering is finished, each tweet in a subevent is associated to its corresponding BMU. Hence, tweets with similar content are distributed in adjacent regions, thereby creating topic clusters.

4.3.2 Creating Stable SOMs

Tracking topical changes in a social stream is regarded as an important task for social media analytics [37] (C1 in Section 5.1). Our proposed SOM method provides a solid foundation for this task. An intuitive solution is to create a series of SOMs that a user can compare and track to understand the topical changes in a social stream. However, original SOM algorithms cannot ensure the dynamic stability of the topics in the maps, rendering the comparing and tracking tasks quite difficult. The random initialization of the weight vectors of the neurons produces topic maps with randomly distributed topic clusters. To solve this problem, we reuse the weight vectors of the neurons in the previous topic map as the initial estimate of the weight vectors of the neurons in the present topic map. This method can largely maintain stable topic maps for adjacent subevents.

Another issue in creating a series of topic maps is ensuring that a consistent global word dictionary is used for building tweet vectors, whose dimensions represent the same words across different maps. However, having such a dictionary is difficult, if not impossible, because a social stream is highly dynamic and the words used in the tweets can be difficult to predict in advance. Gradually building the dictionary is time-consuming and space inefficient. To handle this problem, we introduce feature hashing [46], also known as the hashing trick. This fast and space-efficient method maps arbitrary features (i.e., words in tweets) to indices in a vector. The method applies a hash function to the features and then uses the hash values directly as feature indices of the vector. Therefore, a vector of fixed-length can be easily built. With the method, vectors of the same dimension across different subevents can be used to generate stable topic maps.

4.3.3 Refining SOMs

StreamExplorer allows users to iteratively refine SOMs by merging clusters or by splitting a cluster. When two similar clusters are identified, users can simply select one cluster and drag it to the other cluster area. The system automatically selects the tweets of the source cluster, and the selected tweets are then assigned to new neurons in the target cluster by running the SOM algorithm again. In addition, the system can split a large cluster into smaller clusters using a map in the cluster region with a higher resolution.

5 Visualization Techniques

This section presents the design goals for StreamExplorer, followed by the visual design and interactions.

5.1 Design Considerations

To design StreamExplorer, we held interviews and discussion sessions with six data analysts from universities, including undergraduate students, graduate students, and professors. The participants are not the co-authors of this paper, and they have background in computer science and communication and media studies. They are familiar with at least one analysis tool, such as SPSS and R. They track, analyze, and collect Twitter data for their research or course projects. Most of them know basic visualizations, such as line and bar charts. The discussion sessions intend to understand how the analysts track and explore a social stream (Twitter). We derived a set of design requirements from their feedback and from the knowledge we gained from literature review. The design requirements have been further refined by a series of follow-up discussions with the participants.

T Real-time Tracking of a social stream.

- T1 *Highlighting critical periods from a live social stream.* Constantly tracking a social stream can easily overwhelm users. Therefore, the system should automatically detect and highlight critical periods (subevents) that require considerable attention [37].
- T2 *Displaying multi-facet overview of subevents.* The system should provide a multi-faceted overview of a subevent. This information can help analysts determine which subevents are worthy of further analysis.
- T3 *Revealing the ebb and flow of a social stream.* The social stream should be displayed in the context of recent developments [28]. Animated changes in visualizations can indicate the changes of the social stream that are caused by incoming messages [17].

E Multi-perspective Exploration of subevents.

- E1 *Reviewing past critical periods.* A social stream can produce many subevents quickly, thus increasing the likelihood of users missing several of these. The system should allow users to review past subevents.
- E2 *Summarizing a large volume of microblog messages.* A topical, visual summary of selected messages should

be provided so that users can quickly identify the ongoing topics [8], [49]. The visual summary also serves as a starting point for multi-level visualization [13].

E3 *Allowing for in-depth exploration of the four Ws (who, what, where, and when).* The four Ws regarding an event are considered as the basic elements of information-gathering [45]. The system should help explore the four Ws of any subevent.

C Visual Comparison of subevents and topics.

C1 *Showing dynamic topical and geospatial changes.* Tracking topical and geospatial changes with respect to an event is critical for many applications. The system should properly visualize the changes [17], [49].

C2 *Supporting comparative analysis.* Comparative analysis is strongly demanded in visual text analytics [15]. The system should help investigate the similarities and differences among several topics being discussed.

5.2 Visualization Design

Figure 1 shows our user interface with two major views: a novel timeline visualization for displaying the dynamic changes in tweet volume (upper part in Figure 1 for **T1-T3** and **E1**), and a map visualization for exploring the subevents from the geographic or topical aspect (lower part in Figure 1 for **E2-E3** and **C1-C2**). The interactive lenses allows for further exploration of the four Ws (**E3**) of a subevent.

5.2.1 *Timeline Visualization*

Our timeline visualization consists of two components, namely, trend visualization and sedimentation visualization. We use a DICON to show the profile of a subevent.

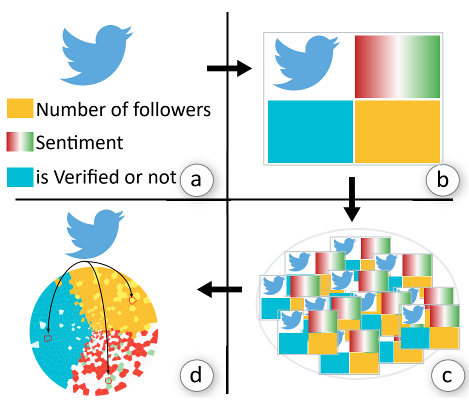


Fig. 3. The process of generating a DICON. (a) A tweet with three properties. (b) Each property is encoded by a cell. (c) Cells are packed to generate an icon. (d) Icons are grouped.

Trend Visualization. The right part of Figure 1(b) is an animated line chart that displays the tweet volume regarding an event over time to reveal the ebb and flow of the event (**T3**). A user can choose different time units (e.g., second, minute, and hour) on the left of the timeline visualization. The subevents detected by the streaming algorithm (Section 4.2) are highlighted to relieve the burden of the monitoring task. This is done by using a DICON in the trend visualization.

DICON Visualization. In accordance with the suggestions of our end users, critical periods should be highlighted from a social stream (**T1**). Meanwhile, a multifaceted overview of the tweets during a critical period should be displayed (**T2**). Highlighting critical periods can be achieved easily by changing the background color of the time periods. Multidimensional data can be displayed using many methods, such as parallel coordinate and scatter plots. However, the space and context of trend visualization constrain us from using these methods.

DICON [11] is an icon-based compact method that visualizes multi-dimensional clusters. Figure 3 shows the process of generating a DICON. DICON helps illustrate both the overall data distribution (the pie chart) and the individual properties (the small cells inside each pie slice). Each cell indicates a property of a data instance, whose size and color double encode the property value. Therefore, for those instances (i.e., tweets) that have very small values in certain properties, the corresponding cells are expected to have a very small size and a very light color. The other cells with larger property values will be highlighted. The layout of the cells in each pie slice is also optimized, such that the (relative) positions of the cells of the same data instance are laid out at the similar places inside each cell. This helps users reconnect the splitting pieces together.

We selected three properties (i.e., *number of followers, sentiment of tweet, whether author is verified or not*) to generate DICONs. *Number of followers* and *whether author is verified or not* are fetched via the Twitter streaming API. The sentiment of a tweet is calculated using a widely-used sentiment analysis tool provided by Stanford CoreNLP [36]. Moreover, this approach is efficient in handling streaming data. The accuracy of the method is higher than 80% [1].

We choose only a part of tweets of a subevent to generate the diagram for two reasons. First, social streams usually contain much irrelevant and noisy information. Visualizing all tweets of all users may obscure the important and relevant information. Second, filtering tweets can reduce the number of tweets to be processed by the system. Prior research reveals that influential users play a gate-keeping role in spreading information [47]. People tend to trust the content produced by influential users more than other sources [23]. Thus, the proposed system uses the tweets of top influential users, which are identified by a well-established method called Klout [1], to generate DICONs to provide concise and informative overviews of subevents. After experiments, we found that top 10% influential users can help achieve a good balance between time performance and coverage quality.

Sedimentation Visualization. As suggested by our end users, the system should reveal the dynamic changes of a social stream in the context of its past evolution (**T3**). The system should also allow them to easily review past significant subevents (**E1**). The past subevents will eventually fade out in a traditional line chart, which displays streaming data such as stock prices. A list of past subevents can be provided to users in another list view. However, this solution may create a gap between the line chart and the list view. Hence, we use the metaphor of visual sedimentation [25] to main-

1. https://nlp.stanford.edu/sentiment/

tain a consistent mental model. This is achieved by keeping aging subevents visible by aggregating them into strata over time. The metaphor is intuitive and can clearly preserve the chronological order of incoming data while avoiding clutter, with smooth transition between incoming and aging data.

A linear visualization similar to SendiClock [25] can be used to display the sedimentation process by aggregating subevents into compact layers. Longer time units are encoded as strata. However, after demonstrating a prototype of this visualization to end users, we identified the following limitations of this method. First, users feel confused about the layers, which are dynamically changing to reserve space for more incoming subevents (tokens). The change in the size of a layer may convey the wrong impression of the change of tokens contained in that layer. Second, the past subevents from the layers are difficult to select, review, and explore.

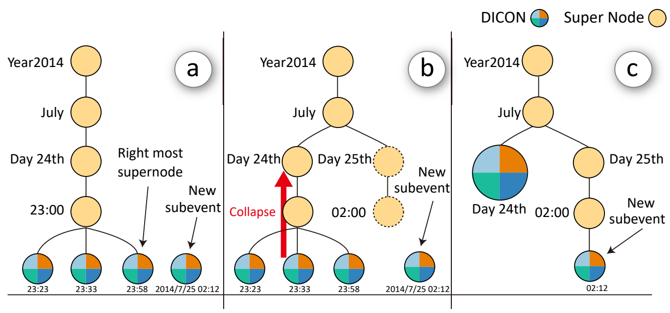


Fig. 4. (a) A new subevent arrives. (b) The node of Day 24th is collapsed, and the two super nodes of Day 25th and 02:00 are added. (c) The new subevent is added as the rightmost leaf of the tree.

To address the problems, we design a novel visualization using a tree (Figure 1(a) and (Figure 4) to show the sedimentation in a hierarchical manner. The visualization can also circumvent the problem of size confusion using the collapse operations of nodes to reserve space for incoming tokens. The hierarchical representation can easily scale up to handle a large number of subevents, thus allowing for the intuitive navigation and exploration of the past subevents.

A tree node represents a layer of the strata (e.g., the set of subevents) in a time range (e.g., week, day, hour), and a tree edge implies a hierarchical relationship (e.g., a minute node includes a set of second nodes). A higher node in the tree encodes a more coarse-grained layer of aggregated subevents. The brown nodes denote super nodes with a fixed size and all its child nodes expanded, whereas the DICON nodes represent regular nodes with their sizes encoding the numbers of subevents within the nodes. A ring is concentric with a DICON node, whose length indicates the mean of the diversity magnitude of the subevent.

We use an algorithm to build and maintain the tree. Figure 4 illustrates the process of adding a subevent to the tree. An aging subevent (token) enters the sedimentation visualization from the entry point in the left border of the changing line chart (Figure 4 (a)). The algorithm traces down the current tree along a path from its root toward its rightmost leaf. If the time range of a node that is currently being visited does not cover the time of the token, then the current node is collapsed (Figure 4 (b)) and becomes a regular node (see the blue node of Day 24 in Figure 4 (c)). A new node is then added to the tree as the rightmost leaf node. Meanwhile, the algorithm continues to handle two cases. In the first case, the time unit of the current node (e.g., minute) is only one level higher than that of the subevent (e.g., second). A new node representing the subevent is added to be the rightmost child of the current node and the algorithm ends. In the second case, the time unit of the current node (e.g., day) is significantly larger than that of the subevent (e.g., second). A series of super nodes are recursively added as the rightmost nodes of the tree (see the brown nodes with the dashed borders in Figure 4 (b)) until the condition of the first case is met.

The visual sedimentation provides users with a quick overview of the past subevent distribution (E1). When a user clicks a node, it is expanded to reveal its child nodes. By recursively doing this, the user can quickly locate a subevent (i.e., a leaf of the tree).

5.2.2 Map Visualization

For further exploration of the subevent (E2, E3, C1, and C2), a map visualization is created and shown to a user (Figures 1 (c)-(e)) when a subevent is selected from the timeline visualization. The map metaphor can visually summarize structures and clustering information in different kinds of data, such as graphs [19] and text [39]. Two types of maps can be chosen by users: an ordinary geographic map showing the geolocations of the related tweets or users and a topic map summarizing the content of the tweets on the subevents. On the one hand, the geographic map can help users understand the distribution of the tweets or the users posting the tweets about the subevent on the map. On the other hand, the topic map presents a visual summary of the tweet content by using the SOM algorithm (Section 4.3). We use the heatmap method to draw both maps, in which the darker regions represent those with more users or tweets. Maps are commonly found everywhere and are intuitive for users to interpret and understand.

Keyword Selection. We place important keywords on the topic map to improve its readability (Figures 1(c)-(e)). To ensure the expressiveness and informativeness of the keywords, we employ a method similar to TF-IDF [30] in order to measure the goodness of the keywords for each map unit. The topic map with the selected keywords allows users to quickly identify the main topics of the subevent without reading the underlying tweets (E2).

Topic Tracking. Understanding the evolution of topics in a social stream (C1) is an important task. Stacked graphs are widely used to track and visualize temporal variations of topics. However, stacked graphs cannot reveal the topology (or relationship) of the topics. In communication research, researchers want to track the evolution of individual topics over time as well as detect and understand any change in the topic topology. To overcome this issue, we allow a user to choose multiple subevents from the timeline visualization and create a series of topic maps for each subevent (Figures 8(f)-(h)). The stable and consistent topic maps (Section4.3.2) preserve the topic topology to facilitate visual tracking and comparison of topics.

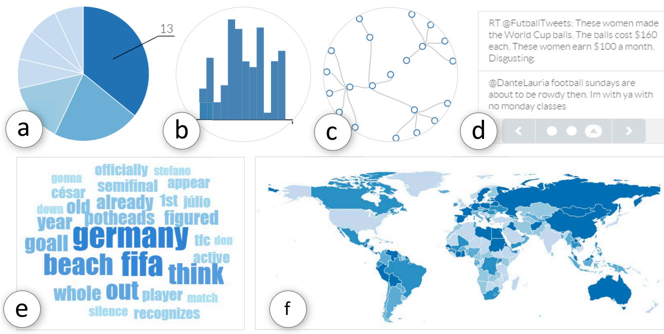


Fig. 5. Interactive lenses: (a) pie lens, (b) bar lens, (c) word lens, (d) list lens, (e) network lens, (f) Geomap lens.

5.2.3 *Interactive Lenses for Multi-faceted Visualization*

Multi-faceted analysis is highly desired, because users want the system to support the interactive exploration of the four Ws (E3) in an intuitive manner using familiar visual encoding or metaphor. Therefore, the system uses map visualization as a basis and employs interactive lenses to facilitate multi-faceted exploration in the map. The metaphors of maps and lenses are commonly found in daily lives. Thus, they are intuitive for our end users to understand and use.

Interactive lenses (or magic lenses) are typically used in data visualization to provide alternative visual representations of the data in selected local areas. Figure 1(f) shows a panel from which users can choose the data dimensions, such as keywords, for further exploration. When a data dimension is selected, a set of visual icons representing different types of lenses is shown under the selected data dimension. The users can drag any visual icon to an area in a map. An interactive lens is then shown to provide an alternative visual representation for the region, thus enabling the user to see and examine the data in the given area from a certain perspective (i.e., selected data dimension). The system currently supports several types of interactive lenses (Figure 5), which are detailed below.

* **Pie lens** provides a visual summary of the data distribution by using a pie chart for a data dimension.
* **Bar lens** is similar to the pie lens in providing an overview of the data distribution but uses a rectangular bar chart.
* **Word lens** presents a word cloud that visually summarizes the keywords contained in any selected map area.
* **List lens** simply shows a detailed list of data items such as users or tweets in any selected map area.
* **Network lens** displays the network structure of the retweet network in any selected map area.
* **GeoMap lens** displays the geolocations of the tweets or users in a topic map area.

With the interactive lenses, users can readily compare different areas in a topic or geographic map from different views (C2). The users simply need to drag the lenses of a certain type (e.g., word lenses) to the areas for side-by-side comparative analysis.

5.3 User Interactions

In addition to basic interactions, such as pan and zoom, several other interactions are supported by StreamExplorer.

Adjusting visual activity. Users can adjust the level of visual activity of the system by changing the time granularity (Figure 1(g)). For example, if tracking a social stream in seconds results in visualizations that change excessively fast, then users can choose the minute granularity.

Refining topic maps. The SOM algorithm can produce reasonable and interpretable topic maps in most cases. However, some scenarios continue to require end users to refine the topics. This function is important for advanced users who need to perform in-depth, rigorous studies and publish the results for a wide audience. The system currently allows users to merge or split topic clusters by brushing and dragging related areas (see Section 4.3.3 for details).

Tracking topical changes. Users can simply choose a topic map and then use it as a classifier when tracking the topical changes. When a social stream has been followed for a sufficient time period, the most recent topic map can cover majority of the topics that have been discussed thus far in this social stream. Using a map as a classifier is beneficial for users for two reasons. First, the system can generate more stable maps that are significantly easier to follow and understand. Second, the performance of the system is improved using the classifier instead of the SOM algorithm.

6 Implementation

Our system has two separate parts from the implementation viewpoint. One part is responsible for computing and storing streaming data on a server, and the other part is responsible for visualizing the data as a web browser application on a client. The client program was developed using HTML5 and JavaScript. We also employed a widely-used JavaScript visualization library, namely, D3.js, to create visualizations in clients. We developed the server program using ASP.NET, and used SignalR to add real-time web functionality to our application. A social stream is handled in the server program by running a subevent detection algorithm, which is implemented by C#, and storing the data to a Microsoft SQL Server. The batch-type SOM method (see Section 4.3) is naturally parallel [27]. Thus, we used NVIDIA CUDA to implement the algorithm and accelerate its performance. However, implementing the method using CUDA was quite challenging, because the vector of a tweet could be very high dimensional. To address this issue, we used a dimensionality reduction algorithm (i.e., Random Mapping), commonly used by SOM methods, to reduce the dimension of the tweet vectors [26].

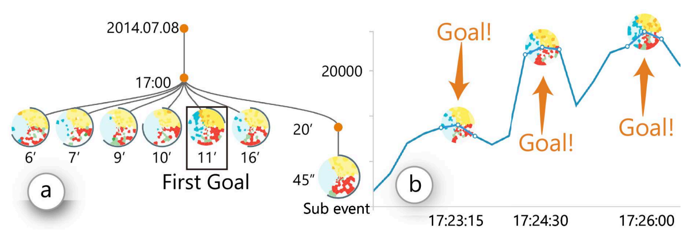


Fig. 6. Timeline visualization: (a) a visual tree of aging subevents, in which Node 11 appears abnormal with the shortest ring; (b) a line chart with three highlighted subevents regarding three goals.

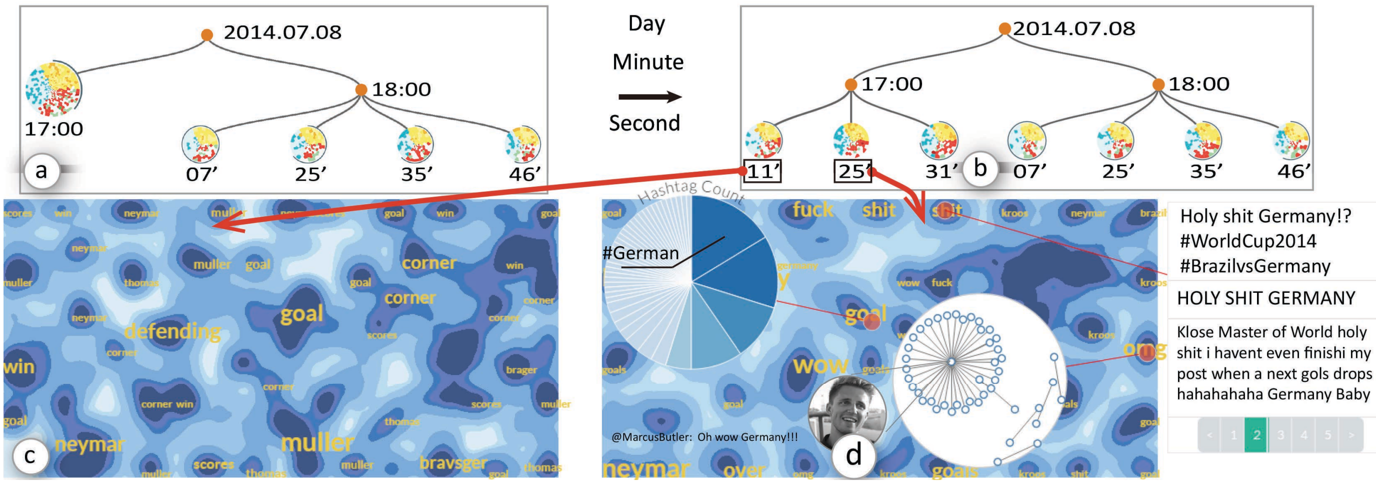


Fig. 7. Case study of the "Brazil 1-7 Germany" football match: (a) visual tree with the time unit of minute; (b) visual tree created by expanding the node of "17:00"; (c)-(d) two topic maps corresponding to the subevents of the node of "11" and "25," respectively.

7 Results

This section presents three case studies with respect to the 2014 FIFA World Cup, the Ebola outbreak in 2014, and the Brexit event in 2016 to demonstrate the use of our system. The FIFA data set contained around 100 million tweets collected using 823 keywords, such as fifa, worldcup, #gre, and #bra, from June 10, 2014 to July 16, 2014. The Ebola data set contained around 20 million tweets with the keyword "ebola" from January 1 to December 12, 2014. The Brexit data set contains around 271,000 geo-referenced tweets with the keyword "brexit" from June 23 to June 25, 2016.

We evaluated the event detection algorithm with precision and recall. Two researchers collected ground truth events and only the events which they both agreed were retained. The ground truth on the soccer game was created based on the online game summaries and game video. For the ground truth on Ebola, we defined the major events as the first announcements of confirmed cases by the respective nation-states, their first deaths, and their first secondary transmissions based on the timeline data recorded by Wikipedia. The ground true events of the soccer game and Ebola cases are concrete and clear. Thus, a researcher coded the events and the other researcher examined the coded events to ensure the validity. The ground truth on Brexit was acquired from the script live broadcast. Brexit is more complex as it covers diverse topics. Therefore, the two researchers coded the Brexit events independently. We measured the inter-coder reliability with Krippendorff's Alpha. The alpha value is 0.832, suggesting that the coding process was reliable.

Regarding the FIFA World Cup, Ebola and Brexit datasets, we collected 22, 10, and 9 ground truth events, respectively. The value of precision and recall were 94% and 72%, 26% and 70%, and 75% and 67% for the above three datasets, respectively. The performance is comparable to that of other existing event detection methods [21].

7.1 Time Performance Analysis

We deployed and tested the system on a laptop with Intel i5-4210M CPU (3M Cache, up to 3.20 GHz), 16GB RAM, 1TB hard disk, and Nvidia GTX 850M GPU (4GB RAM). The client program can run interactively on the laptop. To evaluate the performance of the streaming tweets and detecting subevents, we tested it with batch input of one million tweets. It finished processing the tweets in 158 seconds, so its time performance was 6300 tweets per second on the laptop. Every second, on average, around 6000 tweets are tweeted on Twitter [2] and the public streaming APIs offered by Twitter only return one percent of the requested tweets. In many scenarios, the system only needs to update in every minute or in even longer time. Therefore, the streaming component can efficiently cope with a social stream. The SOM algorithm was executed for a selected subevent on demand and accelerated by the GPU. In the following case studies, the largest subevent can have around 45,000 tweets, which can be clustered by the SOM algorithm in 8 seconds. With our multi-stage framework, the system can achieve interactive performance.

7.2 2014 FIFA World Cup

We conducted the first case study to show the use of our system to track and analyze a social stream. In the case study, the proposed system was used to visualize a social stream with respect to the 2014 FIFA World Cup. We focused on the "Brazil 1-7 Germany" football match that took place on July 8, 2014. The match was the most discussed single sports game ever on Twitter [2]. The timeline visualization in Figure 6 shows the trend of tweet count is updated every 15 seconds (Figure 6(b)) with three recently detected subevents, corresponding to the three goals scored by the German players at 23′, 24′, and 26′, respectively.

The visual tree (Figure 6(a)) shows a visual summary of the distribution of the aging subevents over time. From the tree, node 11 (i.e., the subevent detected at the 11th minute of the match) had the unusually shortest concentric ring compared with other nodes. The short concentric ring encoded the least diversity among the nodes, indicating that the subevent within this node highly concentrated on

2. http://www.internetlivestats.com/twitter-statistics/

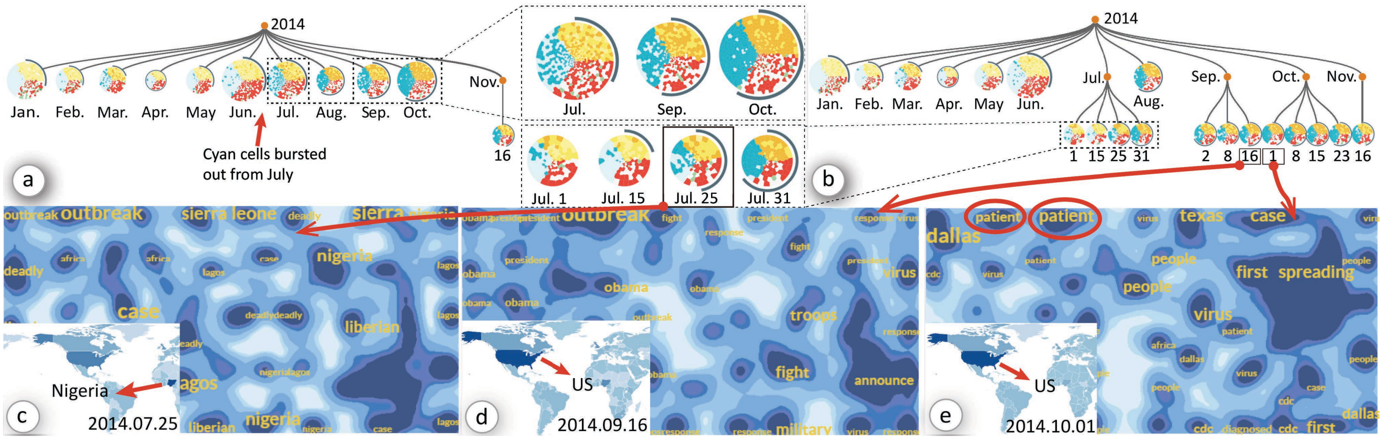


Fig. 8. Case study of the Ebola outbreak: (a) visual tree of the past subevents with respect to the Ebola outbreak in 2014; (b) visual tree created by expanding the nodes of July, September, and October in the original tree shown in (a); (c)-(e) geographic maps showing the geospatial changes of three subevents highlighted in (b) as well as the topic maps showing the topical changes of the three same subevents.

a small number of topics. Furthermore, many cyan cells suddenly emerged in the DICON, implying that many important verified users posted tweets in this subevent. We examined the topic map of the subevent and found that this node contained an important subevent, i.e., the first goal scored by a German player (Thomas Müller). Several keywords, such as “goal” and “mullergoal,” were used intensively and dominated the map, thus producing the least entropy.

Figure 6 illustrates the recently detected and aging subevents that occur in seconds, but the visualization changes too fast to follow. To adjust the visual activity level, we simply changed the time unit from second to minute to provide a higher-level overview of the subevents (Figure 7(a)). We expanded the node of 17:00 in Figure 7(a) and obtained a new tree (Figure 7(b)). Compared with the tree in Figure 6(a), the tree in Figure 7(b) shows less nodes, which consistently corresponded to the major events, such as the goals scored in the match.

As shown in Figure 7(b), node 25 (i.e., the node at 17:25) had the shortest concentric ring and appeared to be more unusual compared with the other tree nodes. Moreover, red cells dominating the bottom right of the DICON indicated the overwhelming negative sentiment in this subevent. Two related topic maps (Figure 7(c) and (d)) were used to compare the topics discussed in nodes 11 and 25. Node 11 covered the time period when the German team scored the first goal, and node 25 covered the time period when three goals were scored by the German team in a short time. Node 11 covered diverse topics, such as “defending,” “Neymar,” “muller,” “goal,” and “corner.” By contrast, the topic map of node 25 exhibited fewer topics denoted by, for example, “wow,” “Neymar,” and “over.” Through the interactive word and list lenses, we found that the topic clusters in node 25 mostly reflected the disappointment of the fans of the Brazil team. As shown by the pie lens in Figure 7(d), “#German” was the most frequently used hashtag in the topic cluster of “goal.” Furthermore, we used a network lens to show the retweet network in the topic cluster denoted by “omg.” The lens showed that “MarcusButler” was the influential user, whose message “Oh, wow Germany!!!” was retweeted by many other users.

This case study proves the usefulness of our system in exploring a social stream. The effectiveness of the subevent detection algorithm and the entropy measure is demonstrated as well.

7.3 Ebola Outbreak

We conducted the second case study to show the usefulness of the system in reviewing and analyzing past subevents. As the news event on Ebola lasted a long time, we used the time unit of days to detect and visualize the subevents.

From the DICON nodes (Figure 8(a)), we found that most of the nodes were occupied by red cells, indicating that a negative sentiment dominated the Ebola outbreak event. We also found that some cyan cells suddenly emerged in July and kept emerging until December, revealing that some important verified Twitter users posted tweets during these periods. Furthermore, although the nodes of July, September, and October exhibited nearly the same size, the yellow cells of September and October are much darker. We then interacted with the tree by expanding the regular nodes of July, September, and October (Figure 8(b)). From the leaf nodes of July, we saw that it was July 25 when the cyan cells emerged. We subsequently created a series of geographic and topic maps (Figures 8(c)-(e)) for subevents to track geospatial and topical changes over time. The sequence of the geographic maps revealed the changes in geolocation distribution of the tweets within the three subevents selected, which occurred on July 25, September 16, and October 1. The maps showed that most tweets came from Nigeria and United States.

To understand why Nigeria and the United States garnered media attention on those dates, we used the topic maps (Figures 8(c)-(e)) for further analysis. Figure 8(c) illustrates some major topic clusters, such as “Nigeria,” “Lagos,” and “Liberia.” Through the interactive lenses, we found an Ebola outbreak in Nigeria caused by an ill traveler from Liberia, who died on July 25. We then studied the remaining topic maps (Figures 8(d) and (e)). The main topic clusters, such as “Obama,” “announce,” “troops,” and “Africa,” (Figure 8(d)) revealed that the corresponding subevent was mainly caused by the heated discussion about Obama’s announcement on September 16 to send 3,000 troops to

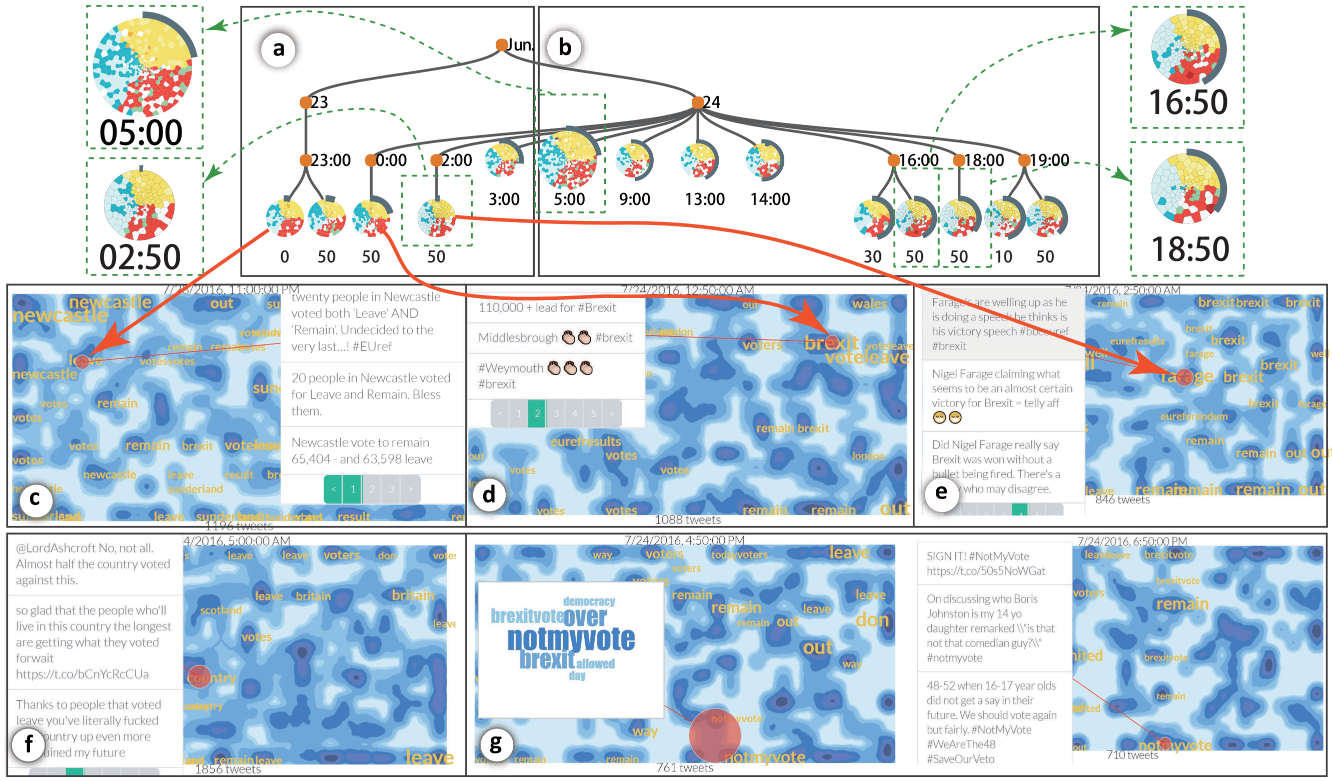


Fig. 9. Case study of the Brexit vote event: (a-b) visual tree of the past subevents; (c-g) topic maps showing the topical changes of corresponding subevents occurred at 23:00 on June 23, and at 00:50, 02:50, 05:00, 16:50, 18:50 on June 24.

fight Ebola in Africa. Figure 8(e) shows some topic clusters, such as “first spreading,” “Dallas,” and “patient,” obviously revealing the first Ebola case diagnosed in the United States. This example proves the usefulness of the maps for a quick analysis of the past subevents.

Our system allows a user to interactively refine a topic map. For instance, we merged the two separate clusters, which are both denoted by “patient” in Figure 8(e) by intuitively dragging the left to the right cluster area. Figure 1(c) presents the refined result. As can be seen, the merged cluster of “patient” appeared to be larger, and the original left cluster was replaced by a new topic cluster denoted by “hospital.” Furthermore, the refined topic map (Figure 1(c)) was used as a classifier to create two topic maps for the subsequent subevents on October 8 and October 15. Figure 1(d) and (e) depict the results, which maintain similar topics consistent and stable on the map. Thus, the generated maps significantly eased the difficulty of visual comparison and tracking of the topic changes among the maps. The topic map in Figure 1(d) shows topic clusters denoted by “Duncan,” “Dallas,” and “died.” These results indicated that the first Ebola patient (Thomas Duncan) diagnosed in the United States died of Ebola in Dallas on October 8. The topic clusters denoted by “Dallas” and “Texas” became larger again in Figure 1(e). Using interactive word and list lenses, we found that these expanded topic clusters were mainly caused by the breaking news that a nurse who came into contact with Thomas Duncan was diagnosed with Ebola in Dallas, Texas.

This case study shows the usefulness of our system in reviewing past subevents. It also shows the use of our SOM method to ensure the stability of a map sequence.

7.4 Brexit Vote

We conducted the third case study to show the effectiveness of our system to track and understand a social stream with DICONS and a sequence of topic maps. Given that the Brexit vote process and discussion lasted around two days, 10 minutes of time unit was chosen to better reflect the dynamics during the event.

The visual tree (Figure 9 (a-b)) presents a visual summary of the distribution of the subevents regarding the voting process. We found that most of the nodes before 05:00 AM of June 24 had relatively shorter concentric rings, indicating that the subevents within these nodes concentrated on a small number of topics. However, the concentric rings of the nodes after 5:00 AM, June 24 were relatively longer, implying that the subevents within these nodes have more diverse topics. We further examined the topic distributions with a series of topic maps for the nodes.

Figure 9 (c-g) presents the topic maps of the first event (nodes of 23:00 on June 23), and subsequent selected events (nodes of 00:50, 02:50, 05:00, 16:50, and 18:50 on June 24). We found that in Figure 9 (c), several keywords, such as “Newcastle” and “remain”, were the most salient among all the keywords. By examining the detailed tweets, we found the vote result of Newcastle to be 51% remain (65,404) and 49% leave (63,598). The votes for remain and leave were rather close, and the topics were basically about discussion and controversy about brexit. However, as time went on, keywords like “Brexit” and “voteleave” started to emerge on the topics maps (Figure 9 (d)). After examining the related tweets, we found that more and more cities’ vote counting had been completed, and the results eventually proved to be “leave” (e.g., *110, 000+ lead for #Brexit*).

In the visual tree, the concentric ring of node 02:50 on June 24 was short, indicating that the discussion highly concentrated on a small number of topics. From the corresponding topic map (see Figure 9 (e)), we found interesting topics of "Farage" and "brexit." Through the interactive word and list lenses, we found that the topic of "Farage" mainly emerged due to the declaration in advance of a Brexit victory by Nigel Farage, the leader of the UK Independence Party, as well as his calls for David Cameron to resign. We also found others' criticisms for Farage's claim of Brexit victory. Furthermore, the node of 5:00 on June 24 was large, and the green cells started to emerge at the bottom right of the DCION, indicating that more users were involved, and that the sentiment appeared to be positive. Moreover, keyword "remain" does not show up on the topic map (see Figure 9 (f)). It seems that the voting result of Brexit had almost been settled at that time period, and users who supported brexit were already tweeting and celebrating.

After the decision of Brexit, we were interested in what topics emerged on Twitter as the concentric rings of the nodes after 5:00 AM of June 24 became longer (Figure 9 (b)). In addition, the number of blue icons referring to the number of verified users decreased, and the number of red cells indicating negative sentiments increased. Thus, we selected the nodes of 16:50 and 18:50 on June 24 to unfold the hidden patterns. From the two topic maps (Figure 9 (g)), the keyword "notmyvote" emerged and started to become popular. An examination of the tweets indicated that the users not only tweeted about the results of Brexit, but also started to complain about it (e.g., *#notmyvote*, *#remain*, *#united*) and called for a revote.

This case study presents the effectiveness of StreamExplorer in understanding the patterns behind a streaming event with a series of consistent topic maps, as well as the overview indication of temporal DICON nodes.

8 User Evaluation

Next, we conducted a laboratory study to examine how well users can use StreamExplorer to track, explore, and gain insights into social streams. The study also aimed to find any usability issues for improving the system.

8.1 Participants and Data

We invited 13 data analysts, including nine male and four female students, to evaluate the proposed system. Six were undergraduate students; seven were graduate students. The participants were recruited from different departments, including Electronic Engineering (2), Mathematics (4), Digital Media (3), Psychology (1), and Agronomy (3). They were not the co-authors of the paper. We pre-screened participants to ensure that they all had experience in analysis. The participants were identified as P1-P13, respectively. The evaluation used the FIFA dataset described in Section 7. The period of a soccer match is relatively short; therefore, it is more suitable to use the dataset for simulating an analysis scenario.

8.2 Tasks and Procedure

In the beginning of each study, we demonstrated every detail of StreamExplorer and gave each participant a tutorial.

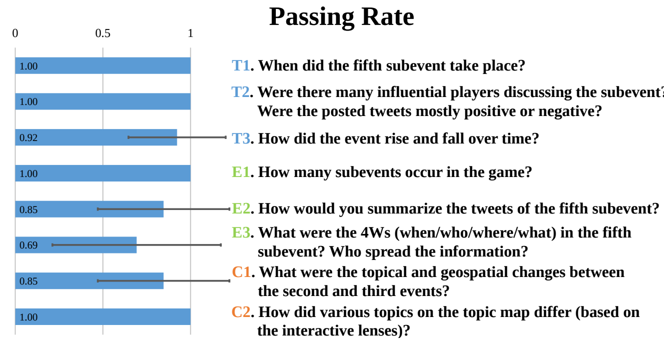


Fig. 10. Evaluation tasks and analysis of the passing rates

Once each participant became familiar with the system, the dataset was loaded, and the tweets were streaming sequentially to simulate a real-world scenario. To ensure the proper length of the study, we streamed the tweets at 3x faster speed. Thus, the 90-minute football match would last 30 minutes. Then the participants were asked to complete a series of tasks (Figure 10) using StreamExplorer to track and explore the social stream. These tasks were classified into three categories corresponding to the design rationale (Figure 10); hence individual visualization components could be properly covered by the tasks. Participants could ask for further explanations of features of StreamExplorer when necessary. The participants were then asked to rate different aspects of StreamExplorer on a Likert scale (1 to 7 ranging from strongly disagree to strongly agree) when they finished the tasks. The task-based evaluation would help the participants better understand the system, thus ensuring that more objective and comprehensive feedback can be elicited from the participants regarding the individual functionalities of the system and the system itself as a whole. At the end of each study, users are allowed to explore the system freely, and we conducted a post-study interview to collect the feedback of the participants. The whole study lasted around 1.5 hours for each participant.

On average, the participants scored 7.3 out of the 8 tasks, and the average passing rate for the tasks was 91.3% with standard deviation of 0.11. The passing rates for T1, T2, E1 and C2 were all 100%, demonstrating the effectiveness of StreamExplorer in tracking the critical time periods, influential users, and the difference of the emerging topics. The passing rate for E3 (*What were the 4Ws in the fifth subevent? Who spread the information?*) was one of the lowest, possibly because the users had not seen such isocontour maps before.

8.3 Results

8.3.1 *Questionnaire*

The participants finished a questionnaire with nine questions. Figure 11 presents the questions and average user ratings. For each question, users were asked to rate their satisfaction regarding three categories, namely, *tracking*, *exploration*, and *comparison*, which corresponded to the design rationale. For example, Q1 corresponded to all three design considerations, whereas Q4 and Q5 (i.e., the map visualization) only corresponded to latter two design considerations.

As indicated by Q1, StreamExplorer received overall ratings of 6 (*tracking*), 5.9(*exploration*), and 6.1(*comparison*) in

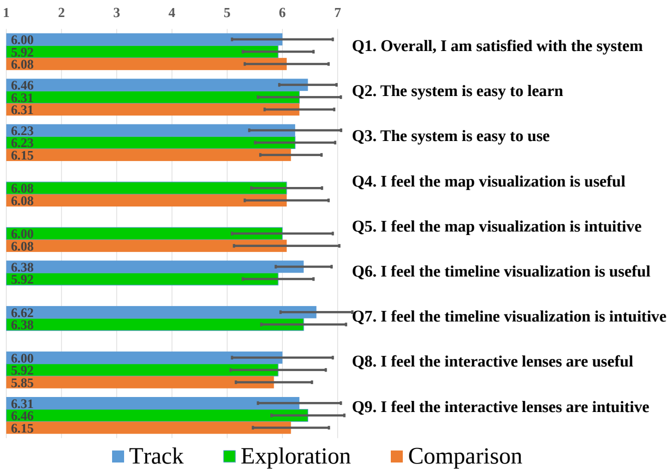


Fig. 11. Analysis of the ratings: all questions received an average rating greater than five, which was very encouraging.

terms of participants' satisfaction. Based on the results of Q2 and Q3, StreamExplorer seemed to be easy to learn and use (all the ratings with respect to the three design considerations were above 6). This is an encouraging outcome, because new users could complete fairly complex exploration tasks with minimum assistance during the experiment.

When considering each component separately, the participants felt that the visual representations and interactions of the timeline visualization and interactive lenses were intuitive during the *exploration* process (Q5, Q7, and Q9). However, even though the map visualization component was thought to be useful (Q4), the participants found the topic map based on isocontours difficult to understand at first glance (Q5). After discussing with the participants, we found that this perception was due to the lack of metaphors in their mind. Nevertheless, after demonstrating the use of topic maps, they became used to it. We also found that the timeline visualization received the highest *tracking* scores (Q6-7) regarding usefulness and intuitiveness. After interviewing the participants, we found that they greatly appreciated the metaphor of visual sedimentation during the *tracking* process. Overall, all questions received an average rating greater than five, which is very encouraging.

8.3.2 User Feedback

System usability. The participants all confirmed that the system was intuitive, clear, and useful. P3 mentioned, *"The tool is great. I can track an event easily and understand the ongoing topics quickly."* His comment touches many of the design considerations of our framework. All the participants appreciated the two key features of subevent real-time tracking and reviewing. A total of 11 out of the 13 participants were impressed by the capability of the system to support the summarization and dynamic topical changes during event exploration. P2 mentioned: *"The feature is extremely useful for understanding topic evolution on social media."*

Visualization design. The visualization design received positive feedback during the interview from the participants. They all agreed that with the help of DICON and topic maps, they could easily navigate from high-level information (e.g., subevents and topics) to low-level information (e.g., keywords and users). All the participants accepted the topic map and wanted to use it in the future. A total of 10 of the 13 participants particularly liked the interactive lenses. According to one postgraduate: "The lens tool is just like real-world lenses. I feel confident in using the tool."

Suggestion. The participants provided valuable suggestion on how to improve the system. Although P3 liked the sedimentation visualization to quickly navigate to a typical subevent, he suggested that the system should also allow him to group subevents based on the contents of the topics. P10 mentioned that he wanted to customize the placement of the properties in DICONs, since sometimes he is only interested in the sentiment distribution. P11 suggested that the system should provide automatic refinement of the keywords on the topic maps, because the duplicated keywords may lead to certain confusion during exploration process.

8.4 Discussion

The results and user evaluation confirm the effectiveness of the system in tracking and exploring streaming microblog messages. Nevertheless, our work has several limitations.

First, StreamExplorer cannot cope with non-English social streams. Our experiments use only English tweets from countries where English is the official language to ensure rigorousness of the experiments. However, relying on a single language may lead analysts to draw biased conclusions because of the restricted representativeness of the message samples. Particularly prone to these issues are countries such as Nigeria where English is the official but not the dominant language. Furthermore, if cellphone and Twitter usage is low, then the data representativeness of social streams can be questionable. In the future, we plan to augment StreamExplorer by incorporating NLP toolkits, such as the Stanford CoreNLP toolkit, to process, track, and visualize non-English social streams. Other sources, such as the news media, can also be considered to expand the coverage of the data.

Second, the performance of the subevent algorithm may depend on the events and the datasets used. For instance, the precision of identifying the major Ebola events is low. The Ebola case lasted a long time and comprised many events. An inspection of the detected events reveals that US users were more active in tweeting and discussing the Ebola outbreak, producing unusually high volumes of tweets (detected as events by our method), than users from other countries. Thus, distinguishing the major events from the other events becomes challenging. Nevertheless, our method achieves a precision of 96% counting the minor events and related discussions.

Third, different clusters may be assigned the same labels because the TFIDF strategy is used to select the labels for the clusters. Therefore, a label may show up several times. We can adopt a strategy that is based on LDA to select the most representative label for a cluster. However, we select the TFIDF strategy to balance the time performance of our system and the representativeness of a label because of the requirement of real-time processing. We provide some interactive lenses to enable users to efficiently explore the map and mitigate the confusion caused by duplicate labels.

9 Conclusion

This study introduces a new efficient framework that incorporates subevent detection and GPU-accelerated SOM for efficiently handling evolving social streams. We further present a multi-level visualization method that integrates a novel glyph-based timeline visualization, a map visualization, and interactive lenses. On the basis of the framework and visualization method, we design and develop StreamExplorer, which empowers users to track, explore, and visualize social streams at different levels on budget PCs. In the future, we will optimize the performance of the system. We also plan to add a new set of interactive lenses, such as the sentiment lens, to the system.

Acknowledgments

The work was supported by National 973 Program of China (2015CB352503), NSFC (U1609217, 61502416, 61602409), the Fundamental Research Funds for Central Universities (2016QNA5014), and the 100 Talents Program of Zhejiang University.

References

[1] Klout. https://klout.com/home.

[2] Twitter data. https://twitter.com/twitterdata/status/486708145775841281.

[3] Special issue in Journal of Communication: Big data in communication research, April 2014.

[4] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao. Twitcident: Fighting fire with information from social web streams. In *Proceedings of International Conference Companion on World Wide Web*, pages 305–308, 2012.

[5] J. Alsakran, Y. Chen, D. Luo, Y. Zhao, J. Yang, W. Dou, and S. Liu. Real-time visualization of streaming text with force-based dynamic system. *IEEE Computer Graphics and Applications*, 32(1):34–45, 2012.

[6] D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, 2012.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

[8] H. Bosch, D. Thom, F. Heimerl, E. Püttmann, S. Koch, R. Krüger, M. Wörner, and T. Ertl. ScatterBlogs2: Real-time monitoring of microblog messages through user-guided filtering. *IEEE Trans. Vis. Comput. Graph.*, 19(12):1077–2626, 2013.

[9] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner. Timelines revisited: A design space and considerations for expressive storytelling. *IEEE Trans. Vis. Comput. Graph.*, To appear.

[10] N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2649–2658, 2012.

[11] N. Cao and D. G. J. S. H. Qu. DICON: Interactive visual analysis of multidimensional clusters. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2581–2590, 2011.

[12] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *Proceedings of IEEE VAST*, pages 143–152, 2012.

[13] J. Chuang, D. Ramage, C. Manning, and J. Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of ACM CHI*, pages 443–452, 2012.

[14] J. A. Cottam, A. Lumsdaine, and C. Weaver. Watch this: A taxonomy for dynamic data visualization. In *Proceedings of IEEE VAST*, pages 193–202, 2012.

[15] N. Diakopoulos, D. Elgesem, A. Salway, A. Zhang, and K. Hofland. Compare clouds: Visualizing text corpora to compare media frames. In *Proceedings of IUI Workshop on Visual Text Analytics*, 2015.

[16] N. Diakopoulos, M. Naaman, and F. Kivran-Swaine. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *Proceedings of IEEE VAST*, pages 115–122, 2010.

[17] M. Dörk, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1129–1138, 2010.

[18] J. Fulda, M. Brehmer, and T. Munzner. TimeLineCurator: Interactive authoring of visual timelines from unstructured text. *IEEE Trans. Vis. Comput. Graph.*, 22(1):300–309, 2016.

[19] E. R. Gansner, Y. Hu, and S. G. Kobourov. GMap: Visualizing graphs and clusters as maps. In *Proceedings of IEEE Pacific Visualization*, pages 201–208, 2010.

[20] E. R. Gansner, Y. Hu, and S. C. North. Interactive visualization of streaming text data with dynamic maps. *Journal of Graph Algorithms and Applications*, 17(4):515–540, 2013.

[21] M. Hasan, M. A. Orgun, and R. Schwitter. A survey on real-time event detection from the twitter data stream. *Journal of Information Science*, To appear.

[22] E. G. Hetzler, V. L. Crow, D. A. Payne, and A. Turner. Turning the bucket of text into a pipe. In *Proceedings of IEEE Symposium on Information Visualization*, pages 89–94, 2005.

[23] M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma. Breaking news on twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2751–2754, 2012.

[24] M. Hu, K. Wongsuphasawat, and J. Stasko. Visualizing social media content with sententree. *IEEE Trans. Vis. Comput. Graph.*, 23(1), 2017.

[25] S. Huron, R. Vuillemot, and J.-D. Fekete. Visual sedimentation. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2446–2455, 2013.

[26] S. Kaski. Dimensionality reduction by random mapping: fast similarity computation for clustering. In *Proceedings of IJCNN*, pages 413–418, 1998.

[27] T. Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998.

[28] M. Krstajic, E. Bertini, and D. A. Keim. CloudLines: Compact display of event episodes in multiple time-series. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2432–2439, 2011.

[29] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the international conference on World Wide Web*, pages 591–600, 2010.

[30] K. Lagus and S. Kaski. Keyword selection method for characterizing text document maps. In *International Conference on Artificial Neural Networks*, pages 371 – 376, 1999.

[31] P. A. Legg, D. H. S. Chung, M. L. Parry, M. W. Jones, R. Long, I. W. Griffiths, and M. Chen. *MatchPad*: Interactive glyph-based visualization for real-time sports performance analysis. *Computer Graphics Forum*, 31(3):1255–1264, 2012.

[32] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. StoryFlow: Tracking the evolution of stories. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2436–2445, 2013.

[33] S. Liu, J. Yin, X. Wang, W. Cui, K. Cao, and J. Pei. Online visual analytics of text streams. *IEEE Trans. Vis. Comput. Graph.*, 22(11):2451–2466, 2016.

[34] Y. Lu, M. Steptoe, S. Burke, H. Wang, J. Tsai, H. Davulcu, D. Montgomery, S. R. Corman, and R. Maciejewski. Exploring evolving media discourse through event cueing. *IEEE Trans. Vis. Comput. Graph.*, 22(1):220–229, 2016.

[35] A. M. MacEachren, A. Jaiswal, A. C. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. Blanford. SensePlace2: Geotwitter analytics support for situational awareness. In *Proceedings of IEEE VAST*, pages 181 – 190, 2011.

[36] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*, pages 55–60, 2014.

[37] A. Marcus, M. Bernstein, O. Badar, D. Karger, S. Madden, and R. Miller. TwitInfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 227–236, 2011.

[38] A. Schulz, B. Schmidt, and T. Strufe. Small-scale incident detection based on microposts. In *Proceedings of the ACM Conference on Hypertext & Social Media*, pages 3–12, 2015.

[39] E. Steiger, B. Resch, and A. Zipf. Exploration of spatiotemporal and semantic clusters of twitter data using unsupervised neural networks. *International Journal of Geographical Information Science*, 30(9):1694–1716, 2016.

[40] G. Stilo and P. Velardi. Efficient temporal mining of micro-blog texts and its application to event discovery. *Data Mining and Knowledge Discovery*, 30(2):372–402, 2016.

[41] G. Sun, Y. Wu, S. Liu, T.-Q. Peng, J. J. H. Zhu, and R. Liang. EvoRiver: Visual analysis of topic coopetition on social media. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1753–1762, 2014.
[42] Y. Tanahashi, C.-H. Hsueh, and K.-L. Ma. An efficient framework for generating storyline visualizations from streaming data. *IEEE Trans. Vis. Comput. Graph.*, 21(6):730–742, 2015.
[43] D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *Proceedings of IEEE Pacific Visualization*, pages 41–48, 2012.
[44] F. Viégas, M. Wattenberg, J. Hebert, G. Borggaard, A. Cichowlas, J. Feinberg, J. Orwant, and C. Wren. Google+ ripples: A native visualization of information flow. In *Proceedings of the international conference on World Wide Web*, pages 1389–1398, 2013.
[45] X. Wang, W. Dou, W. Ribarsky, D. Skau, and M. X. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *Proceedings of IEEE VAST*, pages 93–102, 2012.
[46] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of ICML*, pages 1113–1120, 2009.
[47] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In *Proceedings of the International Conference on World Wide Web*, pages 705–714, 2011.
[48] Y. Wu, N. Cao, D. Gotz, Y.-P. Tan, and D. A. Keim. A survey on visual analytics of social media data. *IEEE Trans. Vis. Comput. Graph.*, 18(11):2135–2148, 2016.
[49] Y. Wu, S. Liu, k. Yan, M. Liu, and F. Wu. OpinionFlow: Visual analysis of opinion diffusion on social media. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1763–1772, 2014.

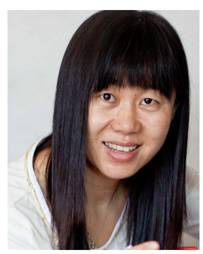
Yingcai Wu is a ZJU100 Young Professor at the State Key Lab of CAD&CG, Zhejiang University. His main research interests are in visual analytics and information visualization, with focuses on user behavior analysis, urban informatics, social media analysis, and text visualization. He received his Ph.D. degree in Computer Science from the Hong Kong University of Science and Technology. Prior to his current position, Dr. Wu was a postdoctoral researcher in the University of California, Davis from 2010 to 2012, and a researcher in Microsoft Research Asia from 2012 to 2015. For more information, please visit http://www.ycwu.org.

Zhutian Chen received the bachelor's degree from the Department of Software Engineering, South China University of Technology in 2015. He is currently working toward the Ph.D. degree in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include immersive analytics, and visual analytics.

Guodao Sun is an assistant professor at the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. He received the B.Sc. in computer science and technology, and Ph.D. degree in control science and engineering both from Zhejiang University of Technology. His main research interests are urban visualization and visual analytics of social media. See more information: http://godoorsun.org.

Xiao Xie received his BS in software engineering from the Sun Yat-sen University, China in 2015. He is currently a MS student in the State Key Lab of CAD&CG, Zhejiang University in China. His research interests are in multimedia visualization and visual analytics. For more information, please visit http://ssxiexiao.github.io/about/.

Nan Cao is a professor at Tongji University. He is also the founder and director of Intelligent Big Data Visualization (iDV^x) Lab. His research interests include data visualization, visual analysis, and data mining. He creates novel visual analysis techniques for supporting anomaly detection in complex (i.e., big, dynamic, multivariate, heterogeneous, and multi-relational) data.



Shixia Liu is an associate professor at Tsinghua University. Her research interests include visual text analytics, visual social analytics, and text mining. She worked as a research staff member at IBM China Research Lab and a lead researcher at Microsoft Research Asia. She received a B.S. and M.S. from Harbin Institute of Technology, a Ph.D. from Tsinghua University. She is an associate editor of IEEE TVCG.

Weiwei CUI received the BS degree in computer science and technology from Tsinghua University, China, and the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, Hong Kong. He is a lead researcher at Microsoft Research Asia, China. His primary research interests include visualization, with focuses on text, graph, and social media. He is a member of IEEE. For more information, please visit https://www.microsoft.com/en-us/research/people/weiweicu/.