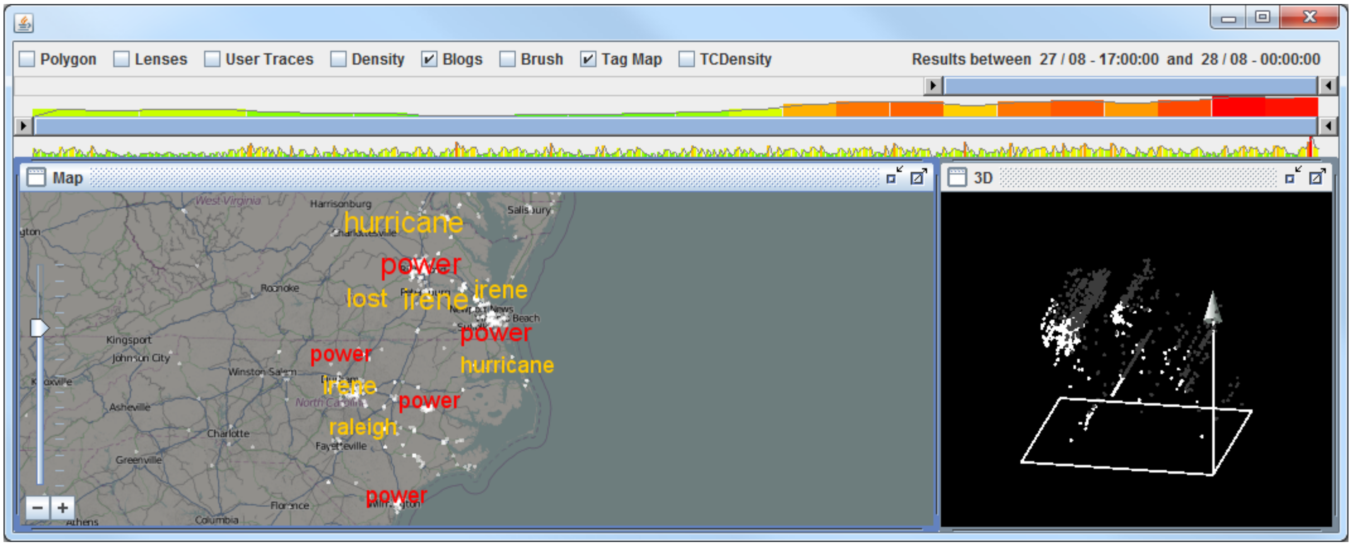
Spatiotemporal Anomaly Detection through Visual Analysis of Geolocated Twitter Messages

Dennis Thom[*] Harald Bosch[†] Steffen Koch[‡] Michael Wörner[§] Thomas Ertl[¶]

Visualization and Interactive Systems Group
University of Stuttgart



☐ Polygon ☐ Lenses ☐ User Traces ☐ Density ☑ Blogs ☐ Brush ☑ Tag Map ☐ TCDensity

Results between 27 / 08 - 17:00:00 and 28 / 08 - 00:00:00

Figure 1: Spatiotemporal anomalies derived from geolocated Twitter messages represented by term clouds on a map. Some of the visible anomalies correspond to power outage events during hurricane Irene on August 27, 2011.

Abstract

Analyzing message streams from social blogging services such as Twitter is a challenging task because of the vast number of documents that are produced daily. At the same time, the availability of geolocated, realtime, and manually created status updates are an invaluable data source for situational awareness scenarios. In this work we present an approach that allows for an interactive analysis of location-based microblog messages in realtime by means of scalable aggregation and geolocated text visualization. For this purpose, we use a novel cluster analysis approach and distinguish between local event reports and global media reaction to detect spatiotemporal anomalies automatically. A workbench allows the scalable visual examination and analysis of messages featuring perspective and semantic layers on a world map representation. Our novel techniques can be used by analysts to classify the presented event candidates and examine them on a global scale.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—GUI; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Clustering

1 Introduction

Users of location-based social or microblogging services can be seen as 'semantic sensors' [6, 20] with the ability to report and describe observations and events in their region by sending short geolocated messages. Therefore, the advent of Location Based Social Networks (LBSN) has brought a valuable data source to analysts in the field of situational awareness and crisis response. MacEachren et al. [16] conducted a qualitative survey amongst emergency management professionals. The result showed that at least 39% of the survey participants already used microblogging tools to gather information from the public and at least 26% used it to monitor the activities of other emergency management personnel.

To improve the process of manually searching and interpreting important messages, geovisual analytics [1] systems can be of great value to the emergency responders. Such systems can provide aggregated overviews that summarize information about a situation and support the interactive exploration of the data.

In analyzing social media data sources, major challenges are how to discover spatiotemporal events and how to handle the sheer volume of incoming data flooding the analysis system. Examples of events, for which significant related social media usage has been noted, include river floods and wildfires [22], earthquakes and typhoons [17, 20], hurricanes [8], technological disasters [23], infectious diseases, and shooting incidents [3, 18, 7]. During such situations, there is often much global and randomly distributed chatter about the event, which is mostly based on public news media or propagated through the social network channels. However, much more valuable and recent information can be gained from people directly participating in or observing the event. In our work we

[*] e-mail: dennis.thom@vis.uni-stuttgart.de
[†] e-mail: harald.bosch@vis.uni-stuttgart.de
[‡] e-mail: steffen.koch@vis.uni-stuttgart.de
[§] e-mail: michael.woerner@vis.uni-stuttgart.de
[¶] e-mail: thomas.ertl@vis.uni-stuttgart.de

build on the assumption that messages from local witnesses of an event can often be detected through their spatiotemporal density, similar term usage, and relative high count of contributing users. In the course of this paper we will therefore define the technical term 'anomaly' to designate a kind of spatiotemporal dense topic clusters where a single term is used in all involved messages. Using this strategy, anomalies corresponding to actual events can already be distinguished from general message clusters originating from high population densities. The task of the analyst within our strategy will be to separate anomalies resulting from minor events or random message groups with identical term usage from such anomalies corresponding to actual important events including crises, catastrophes, and other significant incidents.

To handle the amount of data that necessitates scalable analysis methods and excludes many clustering techniques for anomaly detection, we developed an enhanced Lloyd clustering scheme that is suitable for analyzing microblog posts from a continuous data stream. The clustering algorithm has been optimized for scalability by employing incremental adaption to incoming data. At the same time, it aggregates aging data, while phasing it out of the current clustering state in order to reduce the active, visualized content to a manageable amount. The details of this approach can be found in Section 4.

In this work, we address both of the challenges and build a visual analytics method that is suitable for the detection, visualization, and analysis of spatiotemporal anomalies based on the clustering approach, thereby allowing for scalable aggregation and user interaction. The two main components of the system are an analysis workbench and a monitoring module incorporating the spatiotemporal anomaly detection mechanism. Using the workbench, analysts can review findings, explore the message sets, and capture important events (cf. Section 3). The detection mechanism constantly clusters the incoming data stream on a per term basis and stores spatiotemporal anomalies along with their time and location (cf. Section 4). Since the clustering employed in the detector is designed as a best effort mechanism – being particularly suited for a high performance real time analysis of streaming social media data – the full power of its results unfolds only if it is integrated with an interactive visualization method. Using this method, a broad amount of anomalies can be displayed on a map view using spatiotemporal term clouds aggregating the textual content of messages, which will help analysts decide which anomalies may be important events and therefore qualify for further investigation. Furthermore, interactive query mechanisms can be used to filter or widen the base set of messages, and flexible and adaptive layout mechanisms allow an adjustment of semantic resolution of visualized terms to the zoom level selected in the map-view (cf. Section 4.2).

2 Background

Impact and influence of social media as well as their range of coverage has seen unpreceded growth during recent years. The more information is brought into these media by an increasing number of users, the better they cover and reflect real-world aspects, merely by producing digital traces of them. Often, this helps to communicate events that are interesting for a given task quicker than other channels such as classical news media.

2.1 Geovisualization for Social Media Analysis

The rise of microblogging services and the particular success of Twitter has consequently seen some interest in different research communities trying to exploit these fast new channels, e.g., for building and improving decision making systems for crisis management [24]. However, events or microblog messages, from which events can be derived, are buried under loads of messages that may be irrelevant for an analytic task. Some approaches therefore aim at setting up dedicated social media services with very specific scope and purpose. Examples of such efforts can be seen on the Crisis Mappers Net[1] and in services such as those provided by Ushahidi[2].

In the context of this paper, we concentrate on microblogging services that are not tailored to a specific goal. We use geolocated messages from Twitter as our primary data source. However, our approach is not restricted to data from this specific service. Sakaki et al. [20] also exploit Twitter messages (tweets) for detecting earthquakes and the trajectory of typhoons. Their approach treats Twitter users as social sensors. The users' messages are classified into earthquake related and irrelevant using a support vector machine, taking into account features such as keywords and message word count. They were able to show that specific events can be detected from Twitter messages with high probability if task-tailored models are employed. In their approach, static visualization is used to present the results but not as a part of a visual analytics loop. We follow the idea of detecting a broader spectrum of events by using a detection method that makes much fewer assumptions on the underlying model. We directly exploit messages from Twitter users considering them as semantic sensors.

With Senseplace2, MacEachren et al. [16] presented a prototype system that allows for querying Twitter and depicting aggregated results on a map. The places are determined by employing named entity recognition and reverse geocoding, resolving strings to geolocations. Our approach, in contrast, works directly on the whole stream of geolocated tweets and performs a scalable, geospatial analysis by employing an algorithm in the form of a streaming operator. The study by MacEachren et al. [16] also indicated that the tag cloud view which was suggested as one means for aggregating microblog contents was considered inadequate by some of the specialists. We do not see this result as a contradiction to our approach that uses 'tag clouds' in a broader sense, since the sketched tag cloud of their mockup display, unlike our visualization, is separated from the map view in a layout similar to 'Wordle'. White and Roth presented an application for crime investigation with Twitter-Hitter [27], suggesting geospatial analysis of Twitter messages as a suitable means for addressing different tasks in criminal investigation. Cartoblography by Field and O'Brien [5] discusses the creation of map mashups for representing Twitter content in different layouts.

2.2 Visualizing Information with Tag Clouds

In the context of this document we use the notion 'tag cloud' to refer to a specific type of visual representation. Even if tag clouds are commonly used in web pages today, their meaning, their role in interaction, and their layout differs. They were used to display the most frequent tags assigned by users to digital documents as part of the creation of folksonomies and are now also used to summarize text collections, e.g., by representing the most frequent or important words, and almost any other kind of (meta)data [26]. Furthermore, tag clouds are used as navigation help, e.g., to display a list of electronic documents containing a selected tag, or to facilitate retrieval tasks, such as sending/creating/modifying queries to document repositories as part of a search task [10]. Many layouts exist and provide different benefits for different tasks and situations as, for some of them, described by Lohmann et al. [14]. Multiple new variants of generating and employing tag clouds have been proposed in the InfoVis community during recent years. Wordle[25], for example, has become very popular through its great aesthetics. Also the way that tag clouds could be exploited during analytic tasks beyond providing an abstraction of electronic document collections and allowing for 'topic'-based document navigation has seen some new developments lately. Here, tag clouds are used, as part of other visualizations, for depicting temporal change and trends [4, 11, 21]. If used for showing spatial or spatiotemporal

[1] http://crisismappers.net/

[2] http://www.ushahidi.com/

correlations of digital artifacts in a geographic context, they are also referred to as tag maps [9].

Wood et al. [28] present an approach similar to ours regarding the visual representation of results and the idea of applying spatial clustering. However, their approach differs in some important aspects: They do not exploit the temporal dimension for clustering, which leaves it to the user to explore tag frequency by interactive specification of time intervals, while we employ a tag-cloud-like visualization to represent potentially interesting events in time and space automatically. Furthermore, Wood et al. use a straightforward 'binning' approach, that allows for hierarchical clustering, but comes at the cost of having a fixed, potentially suboptimal grid structure, that might represent spatial clusters inadequately at certain levels of detail. Additionally, our visual approach as well as the employed analytics methods aim at combining realtime and post analysis of events that can be derived from geolocated Twitter microblog messages. We also put special emphasis on visual scalability and scalability of automatic event detection. Trendsmap[3] is another spatiotemporal approach for depicting clusters in geolocated Twitter messages in realtime, but for a fixed predefined time span covering one week. Exploration and analysis of 'historical' data is not possible. To the best of our knowledge there also exists no detailed description on the clustering mechanism as it is used in the Trendsmap tool. The work presented here is quite different from the original idea of tag clouds and has common characteristics with other usages, such as labeling geographical points of interest [15]. However, since users generally refer to our visualization as tag cloud, we will stick to the term.

3 The ScatterBlogs Analysis System

In this section we describe the collection of Twitter data as well as the basics of our visual analytics tool ScatterBlogs. It can be divided into two main components. The first component monitors the incoming Twitter messages and transforms them into a scalable storage format, that supports interactive analysis. The second one is an analysis workbench for the exploration of current and past anomalies that were either detected by the monitor or were defined ad-hoc during the analysis using interactive tools, geographic areas, and terms.

3.1 Monitoring the Twitter Streaming API

Twitter is one of the most popular microblogging services that offers publicly available and geolocated messages of broad coverage. When users post on Twitter, they are aware that anyone on the Internet can access the posting, which means privacy concerns are not a major issue. Most importantly, however, Twitter offers an API to subscribe to a filtered stream of realtime status messages. In addition to author, time stamp, and message text, these contain additional data fields such as other users mentioned, places, links, and potentially the location of origin, either as geo coordinates or as a place name.

According to Twitter, 230 million messages from 100 million active users are processed each day[4]. Of these, we use only messages that come with exact geo coordinates for our analysis: While place names allow for some sort of localization, they are much less precise than geo coordinates and might give the false impression of an amassment of anomalies near a city center. There is no way to locate messages that include neither geo coordinates nor a place name except for a natural language analysis of the message text, which would introduce considerable uncertainty into the localization. This restriction leaves a constant stream of about 1–2 million messages per day. Our prototype uses the free Twitter4J[5] implementation to access the Streaming API service of Twitter. The received information is then stored locally in three different ways. We maintain a database with separate tables for user IDs and messages to allow for on demand access to message details during the analysis. Additionally the message text is fed into an Apache Lucene analyzer[6], which tokenizes the stream into single terms. For now, this supports only English and we discard foreign messages. From the resulting set of terms, we remove web links and frequent stop words and build a full text search index to allow for fast retrieval of message ids for any given term. Once tokenized and cleaned, the term set is passed on to the spatiotemporal anomaly detector along with the timestamp, user ID, and location of the message. The detector creates a list of spatiotemporal term usage anomalies for discovering potentially relevant events. See Section 4 for a detailed description of the spatiotemporal anomaly detector.

3.2 Analysis Workbench

The ScatterBlogs analysis system originated from our VAST Challenge submission [2] for the task of analyzing a large set of synthetic geolocated microblog messages. Its purpose is to enable analysts to work on quantitative as well as qualitative findings by not only automatically identifying anomalies, but also summarizing and labeling event candidates as well as providing interaction mechanisms to examine them.

In order to employ ScatterBlogs in future crisis management or post event analysis scenarios, we upgraded the approach to operate on real life data, with a particular focus on Twitter messages. To achieve this, scalability is a key aspect for most of the components, since we now face a constant high volume data stream, tremendously larger than the synthetic data set for which ScatterBlogs was initially developed. We therefore defined different layers of data examination for an ongoing analysis session to address the most challenging scalability issues. The base layer is the complete dataset of collected messages, which can only be handled in aggregated form in the interactive front end. Nevertheless, the base layer is always available to be included into the current analysis context, which represents the examination layer, and is populated by filtering the base set using full text search or spatiotemporal filtering. The analysis context is transfered into fast storage on demand, preferable to in-memory representations, allowing for an interactive access to the full information of individual messages or message groups.

The most important views of the workbench are an interactive world map, a hierarchical time slider incorporating message density histograms, and a selection management component (for the latter see [10]). All views work on the same data model and show the current analysis context and focus in a synchronized fashion.

The interactive map is used as a display for several data overlays. Here, the individual locations of messages, a heatmap of message densities, and the anomaly indicator presented in this paper can be enabled and disabled. All overlays adjust to the currently focused time, which enables analysts to browse over a time range while watching the change of overlays on the map. The map is also the basis of interactive tools that allow the selection of messages as well as inspecting the details of messages through what we call 'content lens'. The lens samples the most frequent words of the covered messages and displays them in a small tag cloud.This constitutes a very powerful tool for gaining additional insight from an otherwise anonymous message cluster or for browsing over a distributed collection of messages to see if they feature common words.

4 Anomaly Detection and Visualization

In order to select an initial analysis context, as well as to widen an existing context, a technique is needed enabling users to explore the base layer of all recorded messages in an aggregated form. The algorithm described in this paper transforms the stream of geolocated

[3] http://trendsmap.com/

[4] http://business.twitter.com/basics/what-is-twitter/ (on 24th Sep. 2011)

[5] http://twitter4j.org

[6] http://lucene.apache.org

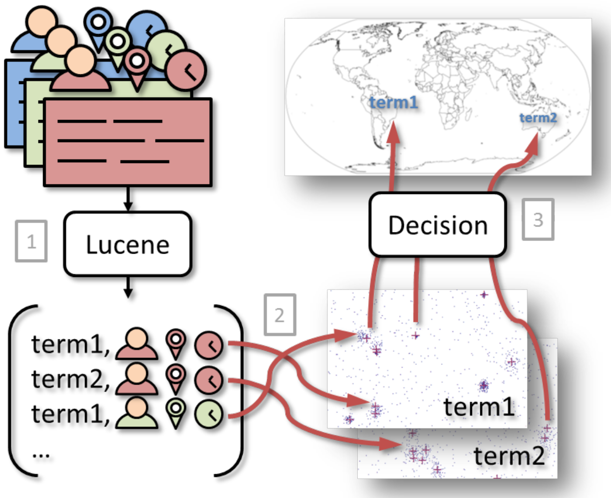


Figure 2: Three activities to generate an overview of anomalies: 1) Message terms are extracted and transformed to term artifacts. 2) Microblog Quantization of term artifacts generates spatiotemporal clusters. 3) Clusters selected by the decision strategy are considered as anomalies and represented as term map overlay for exploration.

messages into a map of terms potentially corresponding to real-life events, which is interactively explorable in both time and space. The algorithm can be separated into three major activities depicted in Figure 2. In ScatterBlogs, two activities are carried out within the Twitter monitoring component while the third one takes place in the analysis workbench.

The first activity is the extraction of terms from messages as they arrive through the Twitter input stream described in Section 3.1. Lucene tools are utilized for tokenizing, stopword removal and handling of special characters. In most cases 5 to 10 terms are extracted per message. Because the remainder of our algorithm works only on individual terms and not their co-occurrence within a message, the output of this activity is a stream of what we call *term artifacts*. Each artifact consists of the term itself, the location and time where it was stated, and the user ID.

The second activity continuously performs a cluster analysis of the extracted term artifacts as they arrive. We developed an enhanced Lloyd scheme [13], which we adapted to detect spatiotemporal clusters of term usage. This component aggregates aging data iteratively, thereby restricting the active amount of data to a size that fits into working memory. The details of this *Microblog Quantization* can be found in Section 4.1. Its result is a set of term usage anomalies. They consist of the term itself, the mean spatiotemporal location of the detected cluster (centroid) and a temporal histogram of the clusters elements.

Based on this data, the third activity generates a graphical representation of the anomalies by placing labels on the map of the ScatterBlogs workbench. The underlying visualization method tackles the problem of showing large numbers of the detected anomalies on an interactive map in a way that the most important anomalies at a given zoom-level are visible. This component will be discussed in Section 4.2.

4.1 Microblog Quantization

The Microblog Quantization has some similarities to the Linde-Buzo-Gray algorithm outlined in [12] as well as the X-Means clustering algorithm [19] but in contrast to these algorithms, it facilitates the scalable and continuous processing of input data. Performing the clustering in realtime and allowing a permanent analysis during an indefinite timespan would be infeasible using common clustering approaches. Firstly, because they are usually not fast enough to do the computation and secondly, because they do not scale to datasets of arbitrary size. With our algorithm we are able to discard old clusters at some point in time and store the location of their representing centroids permanently. This ensures scalability to continuous stream processing of almost arbitrary duration.

To achieve this goal, our clustering algorithm basically performs one global and continuous relaxation step of the regular Lloyd scheme, that is, the clusters are adapted as new messages arrive but globally the relaxation is not done repeatedly until reaching an equilibrium. Instead of using a predefined fixed number of centroids, a splitting scheme is employed to detect new emerging anomalies and also accommodate noise, i.e. term artifacts that do not belong to any actual anomaly. These noise clusters are later discarded after examining the final cluster properties. It is important to note that the algorithm processes the messages on a 'per-term' basis. That is, for each and every term encountered in the messages, a new clustering branch is initialized, which concerns only messages containing this specific term.

The basic algorithm works as follows: We start with a list of empty sets of clusters. As soon as a message is evaluated and its terms extracted, a cluster for every term is generated and a term artifact is assigned to each of them. The centroids of these clusters are initialized with the spatiotemporal location of the message. As further messages arrive, we first check for every term whether clusters for the term already exist and if so, we assign the term artifact to the nearest cluster. We then adjust the centroid of this cluster by calculating the new mean location of the covered term artifacts. Secondly, for every term of the message not already covered by any cluster, we generate a new branch of clusters and initialize them with the corresponding term artifact. Finally, we check for all modified clusters whether the average squared distortion of their elements is below some predefined threshold k. If it is not, we split the cluster into two new clusters and distribute its elements between them using the k-means (in this case 2-means) cluster algorithm.

The following pseudocode for the procedure $handleMsg(T_m)$ illustrates the basic algorithm while omitting the storing and noise cancellation as well as some implementation specific details, which will both be addressed in Section 4.1.1. In regards to the program variables, let T_m be a set of term artifacts that have been extracted from a particular message m. A term artifact t is represented as a record consisting of a term $t.term$, a user ID $t.user$ and a location in a unified spatiotemporal domain $t.loc \in \mathbb{R}^3$. Furthermore let $C(t)$ be a hashmap that uses terms as keys and maps them to sets of clusters. Each cluster c in these sets is a record consisting of its centroid location $c.loc \in \mathbb{R}^3$ and a set of associated term artifacts $c.reg$ called the cluster region.

The procedure is performed for each new message m arriving from the input stream after the term artifacts have been extracted

```
procedure handleMsg($T_m$)
  $Cluster:\ c, c_1, c_2 \leftarrow$ new $Cluster()$
begin
  for $t \in T_m$ loop
    if $C(t.term) = \emptyset$ then
      $c.reg \leftarrow \{t\}$
      $c.loc \leftarrow t.loc$
      $C(t.term) \leftarrow \{c\}$
    else
      $c \leftarrow \arg\min_{\hat{c} \in C(t.term)}(\|\hat{c}.loc - t.loc\|_2)$
      $c.reg \leftarrow c.reg \cup \{t\}$
      $c.loc \leftarrow (1/|c.reg|) * \sum_{\hat{t} \in c.reg} \hat{t}.loc$
      if $D(c) > k$ then
```

```
            $c_1, c_2 \leftarrow split(c)$
            $C(t.term) \leftarrow (C(t.term) \setminus \{c\}) \cup \{c_1, c_2\}$
        end if
    end if
  end loop
end
```

The distortion $D(c)$ of clusters is evaluated using a squared error distortion measure

$$D(c) = \sqrt{\frac{\sum_{\hat{t} \in c.reg} (c.loc - \hat{t}.loc)^2}{|c.reg| * 3}}$$

and the procedure $split(c)$ simply distributes the term artifacts of c between the new clusters c_1 and c_2 and adapts their centroids using a regular k-means algorithm as described in [13]. The threshold k defines the *maximum* expected spatial and temporal extent of a normal distribution in the unified spatiotemporal domain corresponding to what is considered an actual anomaly[7].

Once an anomaly emerges, established clusters should be attracted by the high density of keyword mentions and the closest cluster will absorb the new term artifacts with its centroid moving towards the center of the anomaly. Eventually, the distortion criterion is likely to be met and the attracted cluster is split into one cluster covering the old data and one covering the new anomaly. Using this strategy, we also accommodate and later detect term artifacts not belonging to any actual anomaly. Because this kind of 'noise' will also lead to clusters being split, this strategy prevents clusters from moving too far away from the actual anomaly. The strategy how the noise clusters are detected and discarded is described in the following subsection.

4.1.1 Aging of Centroids and Noise Cancellation

Eventually, a cluster centroid will be so far away from the present time that absorbing a new term artifact will inevitably lead to the cluster being split. When this happens, the resulting cluster covering the old data turns 'stale', meaning that it has no chance of receiving new term artifacts.

At this point, we can chose a strategy to decide whether the cluster represents an actual anomaly and store it in the database and letting it appear in the visualization or if it can just be discarded as noise. We therefore employ a formal definition of 'anomaly' comprised of features that can often be observed in real word events. In this regard, the most important observation is that a cluster will most likely correspond to an actual event if it has both a relatively low average distortion - i.e. densely packed term elements - and at the same time represents a relatively high number of elements. This feature can be expressed with a significance function

$$significance(c) = \sum_{\hat{t} \in c.reg} 1 - \min(1, \frac{\|c.loc - \hat{t}.loc\|_2}{k})$$

Here k has the same value as the threshold we check against $D(c)$. This means that the significance criterion is sensitive to normal distributions above a certain shape and diameter and ignores artifacts that are outside of radius k.

Another feature used in the decision strategy considers the number of distinct users that have been contributing to a cluster and relates it to the total number of contributions. This enables us to exclude clusters which originated from a single user posting the same terms repeatedly, which happens often with automated agents providing weather informations or reporting songs currently played by a radio station.

If the significance of a cluster is very low or if it fails to meet other decision criteria, the cluster likely represents noise and we will discard it once it gets old. In contrast, clusters receiving a high significance rating and meeting all decision criteria are permanently stored and removed from the active computation.

We implement the management of active clusters as well as the discarding of old clusters using a very large FIFO queue of clusters. The cluster sets $C(t)$ of the algorithm in Section 4.1 in fact store just pointers to these representations. Every time a cluster is updated using the $handleMsg(T_m)$ method, the corresponding centroid is removed from its position in the queue and inserted again at the beginning. By simply using a very long queue one can expect that only stale clusters arrive at the end of the queue and once a centroid reaches the end, it is evaluated and then either persistently stored or discarded.

In addition to the centroid position, represented term and significance score of each anomaly cluster, we found it useful to also include additional information that can be used to enhance the visualization. So for each centroid we use the associated messages $c.reg$ to generate a temporal histogram of messages that is divided into equal bin intervals (e.g. hours). To achieve scalability in terms of storage space, we can then discard the regions $c.reg$ of the clusters before transferring them to persistent storage.

To evaluate the performance of this implementation we conducted several tests using a batch analysis of one month of Twitter data. The results showed that on a regular personal computer (Intel Core i7 820QM@1.73GHz and 8 GB RAM) the $handleMsg(T_m)$ method took in average 0.14 milliseconds to process a single message while the overall processing of a message including term artifact extraction took about 0.29 milliseconds. Therefore, with the underlying test setup the system could process roughly about 290M messages per day.

4.2 Scalable Representation by Term Clouds

The output of the clustering component is a spatiotemporal map of anomalies potentially representing actual events. Based on this representation, an overview is generated that represents the anomalies as interactive term clouds on the world map of ScatterBlogs. The label placement algorithm uses the significance and the temporal message histogram computed during cluster analysis to decide on the importance of individual anomalies. Based on these parameters a weight is computed that determines the label size logarithmically. Computing the weights and placing the labels comprises six steps: (i) Filter the anomalies spatiotemporally according to the current viewport and selected time span, (ii) Calculate an initial weight for each anomaly whose temporal histogram overlaps with the boundary of the selected time span, (iii) Determine the desired location and size of the labels based on the computed weights and the centroid locations of the anomalies, (iv) Aggregate the weights of overlapping identical terms and remove the ones with low weights, (v) Sort the labels by weight to guarantee that significant anomalies are placed first, (vi) Incrementally place all labels and try to find the best available position for each.

4.2.1 Basic Term Cloud Layout

In step (i) of the algorithm a filter is applied to reduce the set of relevant anomalies to those located within the chosen time frame and visible map area. This reduction can also be used to display less significant anomalies that would otherwise be dominated by more important ones when selecting larger temporal or spatial frames. However, by considering only the centroids lying precisely within the chosen time range, we would ignore anomalies extending into this range from the outside. To avoid this, the temporal histogram calculated during clustering is used to estimate the amount of messages lying in the intersection. Step (ii) then generates an initial weight for each anomaly by summing the portions of the histogram

[7] For the sake of simplicity we use a single threshold regarding space and time. Therefore the dates and geo coordinates of the original messages have to be mapped to the unified spatiotemporal domain in a way that the maximum extent of anomalies to be detected is covered by the threshold.

covered by the intersection, leading to a value between 0 and 1, and then multiplying this value with the significance of the anomaly.

Based on the weights we have to assign a label, size, and position to each anomaly while addressing the problem of overlapping labels. Especially in zoomed out map views, there is a high probability that large numbers of cluster centroids are located in close proximity, e.g. when different anomalies are co-located at the same city or when people reporting a single event use several different terms, leading to the creation of separate clusters for each term. According to [14], circular tag cloud layouts are suitable to mediate terms in descending order of importance, with the most important label located at the center and the less important ones towards the boundary. In order to avoid occlusion and still convey more information than by only showing the most important anomaly, it seems reasonable to rearrange the less important terms using a narrow circular term cloud centered above the target location. This is achieved in steps (iii), (v), and (vi). For the sake of simplicity, step (iv) is described in the next subsection.

In step (iii) the initial label size is determined by normalizing the computed weight using a logarithmic function along with minimum and maximum size boundaries. After their font size and spatial extent are fixed, we sort the labels according to their weight in step (v).

Finally, in step (vi), we use a layout mechanism described in [15], which allows a high-performance labeling of dense point clouds while at the same time maximizing the utilization of available space: Starting with the most important anomaly, we try to place each label on its desired location. If the required space is already occupied by a previously placed – and thus more important – label, we try to find a nearby vacant area by moving outwards in a circular fashion with a predefined stepping angle and outwards velocity until a free spot is found or a maximum number of iterations is reached. In the latter case, the label will not be placed, because the label position would lie inappropriately far from the location of the anomaly.

4.2.2 Label Aggregation and Semantic Zoom

Since the clustering algorithm is designed as a best effort method, an overfitting of data is likely to occur, at least for anomalies of a certain magnitude. The reason is that the maximum magnitude of the Gaussian distributions we are looking for using threshold k is unlikely to match the one of each and every event. This would both have positive and negative effects. When zooming into the map view, the spatial extent of anomalies can be better represented by multiple identical labels covering the affected area instead of one big label in the middle – e.g. when examining riots taking place in several parts of a city we will see several ‘riot’ labels exactly covering the affected regions. But at the same time this leads to a problem in zoomed out views. If a very important event is distributed over space and time, it is likely to be covered by many centroids having a relatively low significance value. Therefore, each of them could easily be dominated by a less important anomaly possessing a single centroid of relatively high significance score.

We counteract this effect by adding an aggregation step to our layout algorithm. Step (iv) of the algorithm looks for overlapping labels and allows identical labels to supplement each other by accumulating the initial weights, removing the less important one. This way, spatially distributed anomalies have a good chance of constituting a stronger label properly representing the importance of a possible event. Of course, this principle not only applies to the spatial domain but also to any temporal overlap within a selected time frame. If there happens to be one long-lasting continuous event in the same location, it is likely to be covered by multiple clusters distributed over time. But since the corresponding labels would also overlap each other in the spatial domain, they would be aggregated inside the time frame; thus the importance of the anomaly concerning the selected time frame can be properly reflected. The combination of both effects – multiple anomalies per event and aggregated labels – works as a semantic zoom which aggregates anomalies in zoomed out views and splits them to their individual locations when zooming in.

5 Case Study

Our approach is especially useful when complemented with means for interactive inspection and analysis of event candidates. The discussed cases are solely based on actual geolocated Twitter messages recorded during a time period from 08/08/2011 to 09/08/2011. In the following, we describe the exploration of three different events. The first is the earthquake that hit the US east coast region in August 2011. The second scenario comprises the third and following days of the London riots. The last example focuses on hurricane Irene, which differs from the previous two, in that the public was already informed about it through media and US governmental institutions. Due to the prominence of these events, they can be verified through media cross-checks. Such post-event analysis is, of course, relatively straightforward. However, we can illustrate at which point in time an analyst could have detected these incidents using online visual analysis, which would have been of interest at least in the first two cases. Adding monitoring facilities, as suggested in the future work section, could speed up event-recognition time for analysts by guiding their attention through predefined classes of events. All three events are interesting from an analyst’s perspective, who aims at gaining situational awareness. Furthermore, they are interesting since they produce very distinct signals but are still easily detectable with our method.

5.1 Earthquake hitting US east coast

On 23 August 2011 the US east coast was struck by an earthquake with a magnitude of 5.8[8]. This event was very present in Twitter messages, since earthquakes of this intensity are exceptional in the region. Accordingly, our map view represents this event through very prominent labels on different zoom levels. The inspection of Twitter messages on 23 October 2011, either by clicking on one of the earthquake labels or by searching for it using the search box, provides an overview of the places where people were tweeting about it. It is even possible to get a rough idea of the epicenter when browsing through the beginning of the event using our time-slider tool (c.f. Figure 3) at a minute-by-minute resolution. Additionally, the time-slider gives an impression of the number of posted tweets over time, showing the typical distribution of a sudden, unforeseeable event with a steep slope in the beginning and a long tail representing messages echoing the event. Looking at a time[9] period of 6 minutes, it can be seen that the signal gets particularly prominent in areas where the tweet density is higher than in others. We do not see this as a problem, since it is very likely that people tweeting about it in the first minutes have experienced the tremor themselves and are good sources for obtaining on-site information to gain better situational awareness. By clicking on any of the earthquake labels, all Twitter messages, including multiple messages from the same person that contain the key term ‘earthquake’, posted during the set time period are depicted as small red dots in the map view. To take a closer look at these messages, analysts can now use an interactive lens to browse over these messages while getting the gist of them through the display of the most frequent terms under the lens. Reading some of the messages gives the impression that people are upset or shocked, but there seem to be no severe injuries or damage. Adding the search term ‘damage’ to the query and looking for messages containing the term ‘building’ makes this even more obvious (c.f. Figure 3).

[8] http://earthquake.usgs.gov/earthquakes/eqinthenews/2011/se082311a/

[9] All time and date references are given in UTC.

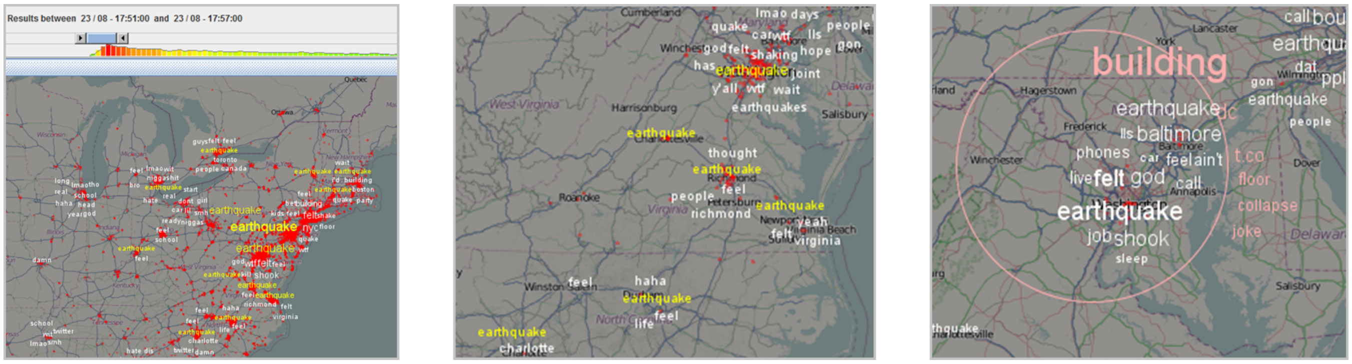


Figure 3: From left to right: The first image shows the map overview showing the earthquake on 23 August 2011. Additionally, the temporal distribution of the term 'earthquake' is depicted in our time-slider tool at a minute-by-minute resolution. The second view depicts all earthquake tags shown within the first minute after the event is mentioned in Twitter. It gives a rough impression of the shock's epicenter, which was actually located 63 miles northwest of Richmond. The third image depicts the usage of our lens-tool while searching for reported damage at buildings.

5.2 London riots

In this second use case, we take a look at the situation in London during the riots[10] on 8 August 2011. The overview shows that a considerable number of tweets refer to riots which is apparent from the very prominent label 'londonriots'. Additionally, the names of some London districts, such as Hackney, Peckham, and Clapham, appear very salient on this day. Again, we drill down by zooming into the region of London, by selecting the label 'riots' to pick our term of interest, and by restricting the time-slider to a range of high message frequency during the evening hours. During zooming, previously combined labels such as 'riot' are split as described and now depict hot spots where the term is mentioned frequently. It is obvious that one of these hot spots indeed lies in the Hackney area. Selecting some of the messages, all sent between 5.30 pm and 6.30 pm, with the lens indicates that the situation was escalating. Police forces seem present at the scene, but the area appears to be an unsafe place at this time:

5.32 pm: Great I,m stuck in the middle of the riots in hackney..

5.37 pm: Road looks to be all fucked up with people and police at bethnal green. #London #riots.

6.07 pm: Just leaving #bethnalgreen it's mad here. Deffo STAY OUT of the area. #riots #London [...]

6.27 pm: Fuck riots in Bethnal green seen fire cabt get home

Similar procedures can be applied to other regions of London, where labels indicate riots and looting, in order to assess the situation there. In the described case, it is quite helpful to have the district names as indicators for where to look for hot spots. Accordingly, we consider keeping such location labels in general but intend to integrate an option to suppress big city labels.

5.3 Hurricane Irene

Hurricane Irene hit the US east coast in the morning of 27 August 2011[11]. In preparation for the storm, some coastal areas were evacuated before it reached the mainland. However, it was unclear how it would affect populated regions in terms of wind speed and rainfall. The analysis is carried out analogous to the other cases. When browsing the time span of interest, it can be seen that several regions are labeled with the term 'power'. Since the term 'power' has ambiguous meanings, it would be interesting to find out if people are referring to power outages and if so, whether these are rather local or regional impacts of the storm. We take a closer look at Richmond, Virgina having several 'power' labels around it. After restricting the set of messages by clicking on one of the labels, we use our lens tool for a more detailed inspection. This shows that the term 'outage' seems to be frequent, too, strengthening the assumption that the power supply might have broken down in some areas. Several parts of the city northwest of Richmond's city center seem to be affected by power outages between August 27th 7 pm to 28th 9 pm, which can be detected by browsing through the filtered messages using the time slider. Determining when power was restored is much more difficult, because far fewer users talk about it. The general assessment of the situation, however, seems to be correct and can be confirmed with information published by the regional power supplier[12].

6 Discussion and Conclusion

In this paper we presented a system that extracts anomalies from geolocated Twitter messages and visualizes them using term clouds on an interactive map. The particular strength of our approach lies in the scalable way of detecting clusters incrementally, which accommodates the dynamic stream of about 1–2M geolocated microblog messages per day. The visualization of anomalies is adjusted to the detection mechanism in a way that scales with the number and diversity of extracted anomalies.

We are aware that usually there is no need to detect natural hazards like earthquakes and exceptional weather phenomena, because there are a variety of institutions using specialized detection equipment and forecast models for this purpose. However, we see two benefits from the analysis of microblogging messages in addition to these secure and reliable sources of information. By analyzing events using Twitter, naturally all events affecting people who are tweeting are integrated within the same approach. The second, more important aspect in case of potentially disastrous events is the possibility to obtain a general impression of how severe the situation is for the people on site. In order to assess the situation, it is important to see where microbloggers are talking about the incident, which in turn can be regarded as a form of event detection, making up the motivation for 'detecting' events. We consider the involvement of human analysts mandatory to reflect and rate the information transported by microblogging, since user statements can be biased regarding over- or understating situations.

Automatic methods, such as our anomaly detection mechanism, always pose the risk of losing important information, e.g, ignoring small spatiotemporal events or overfitting large events. In order to address this issue, we use different strategies. We set the maximum extent of anomalies for our detection mechanism to about the size of a city district and a time period of half a day. This is clearly too

[10] http://en.wikipedia.org/wiki/2011_England_riots

[11] http://en.wikipedia.org/wiki/Hurricane_Irene_%282011%29

[12] http://www.dom.com/storm-center/dominion-electric-outage-map.jsp

small for large events and may be already too large for small ones. Hence to address overfitting, we choose a method for visualizing labels that aggregates the displayed information when zoomed out. Because our clustering is *term-based*, it is very likely that small events covering terms of generally moderate spatiotemporal distribution are captured by a cluster. Instead of setting an explicit threshold for the number of term occurrences in such a case, the significance function implements a tradeoff between number and density thereby boosting small events with high message density. This results in the recognition of local hot spots, as has been described in the analysis of the riots in London, but also enables us to discard noise resulting from broadly scattered messages.

Normalizing term distributions to population densities is not a feasible option here, since it does not correlate well to the distribution of tweets and the terms within them. Exploiting general term density in Twitter messages based on the evaluation of recorded data sets might be a solution that could overcome the discrepancy of population density and Twitter usage. However, it is a very difficult task to achieve this taking into account the quick change and development of Twitter usage worldwide along with seasonal deviations. Moreover, this would require the exclusion of terms related to major events in order to reasonably compute a background noise level (e.g., if there have been a lot of severe earthquake events as has been the case in 2011), which in turn would be used for detecting these events in the first place. Nevertheless, future work will encompass building a general model for spatiotemporal term distribution and its influence on our detection approach regarding accuracy.

Acknowledgements

This work was partially funded by the the German Federal Ministry for Education and Research (BMBF), the German Science Foundation (DFG), and the European Commission as part of the VASA project, the SPP 1335 and the FP7-project PESCaDO (FP7-248594). We would like to thank the reviewers for their valuable suggestions and comments, which helped to improve the presentation of this work.

References

[1] G. Andrienko, N. Andrienko, D. Keim, A. M. MacEachren, and S. Wrobel. Challenging problems of geospatial visual analytics. *Journal of Visual Languages &amp; Computing*, 22(4):251 – 256, 2011.

[2] H. Bosch, D. Thom, M. Wörner, S. Koch, E. Püttmann, D. Jäckle, and T. Ertl. Scatterblogs: Geo-spatial document analysis. In *Visual Analytics Science and Technology, 2011. VAST 2011. IEEE Conference on*, 2011.

[3] C. Chew and G. Eysenbach. Pandemics in the age of Twitter: content analysis of Tweets during the 2009 H1N1 outbreak. *PLoS One*, 5(11), 2010.

[4] C. Collins, F. Viegas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 91 – 98, oct. 2009.

[5] K. Field and J. O'Brien. Cartoblography: Experiments in using and organising the spatial context of micro-blogging. *Transactions in GIS*, 14:5–23, 2010.

[6] M. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.

[7] T. Heverin and L. Zach. Microblogging for crisis communication: Examination of twitter use in response to a 2009 violent crisis in seattle-tacoma, washington area. In *Proceedings of the 7th International ISCRAM Conference–Seattle*, 2010.

[8] A. Hughes and L. Palen. Twitter adoption and use in mass convergence and emergency events. *International Journal of Emergency Management*, 6(3):248–260, 2009.

[9] A. Jaffe, M. Naaman, T. Tassa, and M. Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, MIR '06, pages 89–98, New York, NY, USA, 2006. ACM.

[10] S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative integration of visual insights during scalable patent search and analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 17(5):557 –569, may 2011.

[11] B. Lee, N. Riche, A. Karlson, and S. Carpendale. Sparkclouds: Visualizing trends in tag clouds. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1182 –1189, nov.-dec. 2010.

[12] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *Communications, IEEE Transactions on*, 28(1):84–95, 1980.

[13] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[14] S. Lohmann, J. Ziegler, and L. Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. In T. Gross, J. Gulliksen, P. Kotz, L. Oestreicher, P. Palanque, R. Prates, and M. Winckler, editors, *Human-Computer Interaction INTERACT 2009*, volume 5726 of *Lecture Notes in Computer Science*, pages 392–404. Springer Berlin / Heidelberg, 2009.

[15] M. Luboschik, H. Schumann, and H. Cords. Particle-based labeling: Fast point-feature labeling without obscuring other visual features. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1237–1244, 2008.

[16] A. MacEachren, A. Jaiswal, A. C. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. Blanford. Senseplace2: Geotwitter analytics support for situational awareness. Providence, RI, 2011. IEEE Conference on Visual Analytics Science and Technology.

[17] M. Mendoza, B. Poblete, and C. Castillo. Twitter Under Crisis: Can we trust what we RT? In *Proceedings of the First Workshop on Social Media Analytics*, pages 71–79. ACM, 2010.

[18] L. Palen, S. Vieweg, S. Liu, and A. Hughes. Crisis in a networked world: features of computer-mediated communication in the april 16, 2007, virginia tech event. *Social Science Computer Review*, 2009.

[19] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

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

[21] L. Shi, F. Wei, S. Liu, L. Tan, X. Lian, and M. Zhou. Understanding text corpora with multiple facets. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 99 –106, oct. 2010.

[22] K. Starbird and L. Palen. Pass it on?: Retweeting in mass emergency. In *Proceedings of the 7th International ISCRAM Conference, Seattle, WA*, 2010.

[23] J. Sutton. Twittering tennessee: Distributed networks and collaboration following a technological disaster. In *Proceedings of the 7th International ISCRAM Conference–Seattle*, volume 1, 2010.

[24] B. Tomaszewski, A. Robinson, C. Weaver, M. Stryker, and A. MacEachren. Geovisual analytics and crisis management. In *In Proc. 4th International Information Systems for Crisis Response and Management (ISCRAM) Conference, Delft, the Netherlands*, 2007.

[25] F. Viegas, M. Wattenberg, and J. Feinberg. Participatory visualization with wordle. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1137 –1144, nov.-dec. 2009.

[26] F. B. Viégas and M. Wattenberg. Timelines: Tag clouds and the case for vernacular visualization. *interactions*, 15:49–52, July 2008.

[27] J. J. D. White and R. E. Roth. Twitterhitter: Geovisual analytics for harvesting insight from volunteered geographic information. Zurich, Switzerland, 14-17th September, 2010 2010. GIScience.

[28] J. Wood, J. Dykes, A. Slingsby, and K. Clarke. Interactive visual exploration of a large spatio-temporal dataset: reflections on a geovisualization mashup. *IEEE Transactions on Visualization and Computer Graphics*, pages 1176–1183, 2007.