ScatterBlogs2: Real-Time Monitoring of Microblog Messages Through User-Guided Filtering

Harald Bosch, Dennis Thom, Florian Heimerl, Edwin Püttmann,
Steffen Koch, Robert Krüger, Michael Wörner, and Thomas Ertl

Abstract—The number of microblog posts published daily has reached a level that hampers the effective retrieval of relevant messages, and the amount of information conveyed through services such as Twitter is still increasing. Analysts require new methods for monitoring their topic of interest, dealing with the data volume and its dynamic nature. It is of particular importance to provide situational awareness for decision making in time-critical tasks. Current tools for monitoring microblogs typically filter messages based on user-defined keyword queries and metadata restrictions. Used on their own, such methods can have drawbacks with respect to filter accuracy and adaptability to changes in trends and topic structure. We suggest ScatterBlogs2, a new approach to let analysts build task-tailored message filters in an interactive and visual manner based on recorded messages of well-understood previous events. These message filters include supervised classification and query creation backed by the statistical distribution of terms and their co-occurrences. The created filter methods can be orchestrated and adapted afterwards for interactive, visual real-time monitoring and analysis of microblog feeds. We demonstrate the feasibility of our approach for analyzing the Twitter stream in emergency management scenarios.

Index Terms—Microblog analysis, Twitter, text analytics, social media monitoring, live monitoring, visual analytics, information visualization, filter construction, query construction, text classification

1 INTRODUCTION

Microblog streams published through services such as Twitter contain millions of messages per day. Some of these messages report on the effects of major crises and catastrophes, including natural and man-made ones. Others report on those of smaller events, such as local fires, injured people, or problems during evacuation measures, which can increase situational awareness for crisis response and disaster management tasks. Of course, much noise is transported through these public message channels and even messages of interest have to be handled with care regarding their validity. The latter can hardly be resolved automatically and requires human judgment. But even if the number of microblog posts would be restricted to a smaller geographical region such as a county or a city, human effort alone does not suffice to deal with the enormous volume. The dynamic nature of streaming data, as available from microblog services, makes the analysis even more difficult, since at least some of the above-mentioned tasks require a timely reaction of decision makers.

We therefore propose ScatterBlogs2, a visual analytics [26] approach to facilitate sensemaking [16] of geo-located microblog posts by enabling analysts to create automatic methods for extracting messages and by applying those methods when monitoring topics of interest. Our approach comprises two stages. The first stage lets analysts create classifiers and filters in a visual, interactive way by training and testing them on recorded microblog messages reflecting a priori known situations of interest. The second stage aims at supporting analysts during real-time monitoring of messages that have been extracted using the tools created in the first stage. The interface of the second stage provides interactive visual means for employing and combining the classifiers and filters, as well as for subsequent analysis tasks.

There are a variety of requirements when creating suitable interfaces for both stages. The most important demands for the creation of filter methods are cost effectiveness on the one hand and filter quality on the other. Van Wijk et al. [29] presented their definition of a cost function for visualization that relates users' effort to the value of the insight they draw from a visualization. This applies similarly to the creation of task-specific analytic methods using interactive visualization, which of course includes drawing insight from visualization as an important step. Experience and previous knowledge has influence on the time a user needs to perform the task of classifier and filter creation. Hence, machine learning and retrieval experts as well as analysts that are lay users with respect to these fields, must be enabled to create filters quickly. Accordingly, we developed visual, interactive means for letting users create and test their task-specific filter methods interactively. They can be used to explore and exploit geo-referenced messages for this creation. Even if microblog posts are rather short, we see a benefit in training classifiers and creating statistically backed techniques as opposed to keyword search, because of these techniques' characteristic to generalize beyond an initial set of key terms selected by an analyst. Nevertheless, available keyword queries can be integrated easily with such an approach, by applying them as additional filters. Potentially interesting events in large numbers of microblog messages can be found, e.g., by employing a mechanism to identify spatiotemporally abnormal term usages of multiple persons as has been presented in our previous work [25]. However, one very specific focus of our approach is the detection of relevant low-frequency observations and reports, which are of particular interest for decision makers. Messages containing such information are of much higher value, e.g., during crisis management than the mainstream chatter about topics that are already well-known at the time the majority of users distribute them or comment on them.

In the monitoring stage, we aim at supporting analysts in detecting relevant messages. Here, we suggest another multi-stage process, that optionally lets users define a region of interest, enables them to activate the classifiers that have been created previously, and present the monitored messages as well as their temporal and geo-spatial distribution in real-time. Since it is hard to pay attention to a software tool over an extended period of time (without interruption), methods are introduced that show interesting messages or agglomerations of them over a user-definable period of time. Older but relevant messages can be saved into longterm storage to make them available for training and testing new filters, as well as for improving existing ones.

In the context of this work and our experiments we used microblog posts recorded or streamed from Twitter for filter creation as well as for filter evaluation. Additionally, we tested the trained filters directly

• *All authors are with the Institute for Visualization and Interactive Systems, University of Stuttgart.*
Email: <firstname.lastname>@vis.uni-stuttgart.de.

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

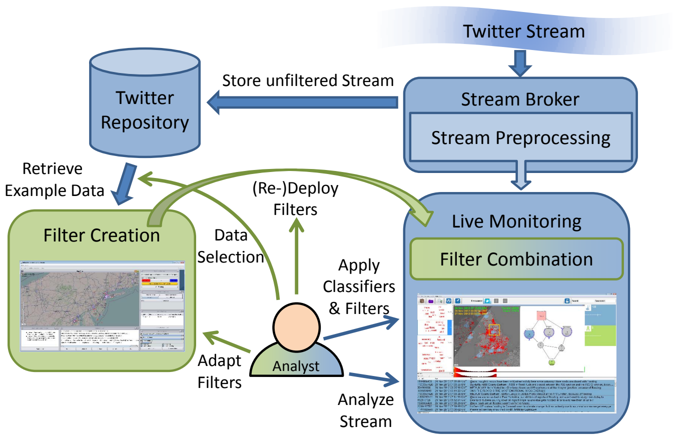

Fig. 1. The stream of geo-located Tweets is stored in a repository, which can be used to train classifiers and to create statistically motivated, weighted keyword filters. These filters can afterwards be exploited during live-monitoring.

on live streaming data. Fig. 1 provides a schematic overview of our approach. We anonymized all message ids and author ids, as well as explicit references to other messages to ensure a minimum of privacy protection[1].

The rest of this paper is structured as follows. The subsequent section sheds light on previous work of filtering and monitoring streamed data in general, as well as on the interactive creation of mechanisms for this purpose. Furthermore, current approaches for monitoring and analyzing geo-referenced documents including microblog posts, are discussed. Sections 3 and 4 provide details on the technical background of our work and describes our main contributions. While Section 3 covers the techniques for filter and classifier creation, Section 4 presents approaches for combining them to achieve situational awareness during live monitoring. Two example usages of our analytic method are described in detail in Section 5. The last two sections present our results, discuss the pros and cons of the approach, summarize our work, and give an outlook on our planned future research activities in the field of microblog analysis.

2 Related Work

Our visual analytics approach touches a variety of scientific fields and technical domains. In the following we concentrate on embedding our approach into related work according to visual filter and classifier creation, interactive document retrieval and microblog analysis, as well as monitoring of dynamic data streams for gaining situational awareness.

2.1 Interactive Filter and Classifier Creation

In 1994, Shneiderman [22] proposed the filter-flow method as a means for creating user-defined queries through applying filters on structured data that could be combined in a Boolean manner. With DataMeadow Elmquist et al. [9] proposed a similar approach, that takes into account multivariate data. Our method of combining filters differs in its adaptation to dynamic data streams and its additional support of advanced filter methods for unstructured data, such as classification methods from the field of machine learning. Similar to VisGets [7], we let users integrate metadata-based filters, i.e., spatiotemporal restrictions and keyword queries, directly from the multiple coordinated view [17] environment. The interface for the creation of advanced filters provides similar mechanisms, since it is beneficial to train and configure methods with both relevant and irrelevant messages.

There are previous approaches that aim at combining human knowledge and machine learning algorithms to support the creation of classification models. This can be done by offering support to efficiently label training data for training algorithms, or by directly giving humans insights or control over the model creation process. Seifert et al. [19] propose an approach that supports users in efficiently labeling training data for document classification by generating a document landscape from unsupervised clustering that enables users to spot regions of documents that can be labeled identically. A similar approach is used by Möhrmann and Heideman [15] in the context of image classification. Their approach is designed to facilitate the labeling of large image datasets. Settles [20] presents a system for creating text document classifiers where both users and the system can take the initiative in labeling or asking for labels, respectively. A classifier creation system that allows users to efficiently label instances and, if necessary, directly influence the classification model created from this data is presented by Hoeferlin et al. [11]. Our previous work [10] compares three approaches to efficiently create training sets for a text document classifier that offer different degrees of control for the user. We found that offering more control to users requires a longer learning phase but offers a higher level of flexibility and control of classifier quality.

Visual analytics approaches have been proposed that provide support for the analysis of text data incorporating automatic text mining components to support users. With Jigsaw, Stasko et al. [23] presented a system that supports the analysis of multiple documents focusing specifically on entities and their relation in those documents. Another system that supports the analysis of documents by presenting them in a landscape metaphor according to their topic is the IN-SPIRE system [33]. Both approaches are well-known representatives for visual text analytics and, of course, there are others as well. To the best of our knowledge, no visual analytics system exits that integrates post-analysis of prerecorded documents to leverage filter creation for real-time monitoring tasks of highly dynamic document streams. Furthermore, we are not aware of visual text analytics approaches for orchestrating keyword filters and classifiers in an ad-hoc fashion, to help analysts in evaluating and improving complex filter mechanisms during online monitoring.

2.2 Microblog Analysis

With the rise of services such as Twitter and Sina Weibo, research regarding the analysis of microblog messages has become a very active field during the last years and plenty of approaches have been suggested to achieve specific analytic goals. Yet, most of the approaches focus on the data analysis of historic Tweets. For example, Sakaki et al. [18] interpret Twitter users as social sensors to successfully determine the epicenter of an earthquake by analyzing delays in messages about the event. Weng et al. [31] find influential Twitter users for specific topics based on an LDA topic analysis of their Tweets. Zhao et al. [35] establish a connection between Twitter and traditional news media by summarizing and categorizing Tweets also based on topic models.

In recent years, researchers have also proposed several visual analytics approaches aiming at real-time microblog analysis that often facilitate interactive means for exploration and anomaly indication. Twitcident [1] uses web-based analysis to connect twitter messages with news feeds from emergency responders, Twitinfo [14] automatically detects and labels unusual bursts in real-time Twitter streams, Leadline [8] connects events in the message stream with recognized entities, Whisper [5] visualizes geosocial information diffusion, and Senseplace 2 [13] provides an integrated geovisualization environment to filter and automatically localize messages and events based on textual content.

To the best of our knowledge none of these approaches aims at combining real-time analysis with interactive classifier training in order to let analysts create task-specific retrieval mechanism that can be orchestrated for the detection of low-profile events or even single important messages in microblog streams. Most of the approaches concentrate on the detection and analysis of high-frequency events. While this can lead to interesting findings with respect to what users of microblog services are talking about most, the information such messages convey is very often ‘second hand’ and can be easily found by other means as

[1] We are aware that microblog messages are a means for achieving broadest possible publicity, but we have reasonable doubt that all users keep this in mind when writing them.

well (e.g., by turning on the news on TV). Consequently, such messages are typically not interesting for gaining situational awareness.

3 Statistically Motivated Filters

In this section, two different techniques are described that enable analysts to create task-specific filter methods for live monitoring. Both methods try to exploit statistical data from recorded messages describing known events in order to define filters that can detect future events of the same type. For this purpose, the unfiltered stream of messages is recorded continuously (see Fig. 1). Whenever a noteworthy event happened, for which filters should be defined, the analyst can select the spatiotemporal extent of the event and persist the messages as a training dataset. Interactive views for explorative analysis support analysts in creating filters that accurately capture relevant data of the events, and at the same time maintain generality over aspects specific to single instances of events, such as geographic names or dates.

The first method can be seen as an extension of common keyword queries and is described in Section 3.1. It facilitates the expansion of a query in a controlled manner through inspecting the statistical distribution and co-occurrence of terms in message sets that describe interesting events and observations. Such a mechanism is useful in cases where an analyst has a starting point for an initial keyword set. Filters constructed with this technique can be used to either widen the monitoring or to suppress certain types of messages.

In some cases it can be hard to encode a specific information need adequately in a keyword query. One reason for these problems is that important low frequency terms will likely go unnoticed by analysts creating the filter, and thus the method described in Section 3.1 can fail. However, a classifier trained on a carefully selected set of training messages, will keep these rare terms in its model, weighting them according to their usefulness to the classification task. Therefore, the second method is a means for creating classifiers and is described in Section 3.2. It facilitates the interactive visual labeling of relevant and irrelevant messages to train SVM based classifiers used as filters.

3.1 Query Widening and Keyword based Filters

In an analysis and monitoring situation analysts are frequently in need of a very good weighted keyword list covering a large set of potentially relevant documents. Especially in the context of social media analysis, it can be difficult to identify such keywords manually, as term usage in social media can differ from analysts' expectations. Therefore, we suggest using messages concerning past incidents as an indicator for term usage of similar future events. While exploring a past event, the analyst may find new interesting facets of it and thereby new relevant terms to be included.

Thus, one first has to find a representative set of messages that describes the recorded incident well in order to identify characteristic terms. As a starting point, temporal and geo-spatial borders of a past event can be exploited to extract a set containing the relevant messages from an unfiltered message collection. In addition, the data can be roughly filtered thematically using regular keyword queries.

Based on this set, the terms most specific to the event are found by comparing their term frequency within the concatenated messages of the relevant set against their inverse document frequency within a considerably larger document corpus, e.g., taken from the same geographic region without a keyword filter. Subsequently, the terms are ranked according to the resulting weight and the top terms are considered for further evaluation. Table 1 illustrates common weight values for a dataset compiled of messages published during the hurricanes Sandy and Irene. Here, one can observe that while all terms are inherently related to the hurricanes due to the initial query, only the top ranked terms are useful for finding additional messages on this topic and evaluating their relevance. Using two separate instances of hurricanes events helps in reducing the weight of non-generic terms such as the name of the hurricane, because Sandy was probably not included in any Irene related message. Nevertheless, `irene` was prominent enough to score a high weight and the analyst should not be considering this term for further evaluation.

As the extracted terms of social media messages will still contain many unusual terms specific to the observed event but not relevant in a generic keyword list, such as names or neologisms, these type of terms are filtered by using a dictionary. To restrict the influence of this dictionary, we only remove terms if their frequency is below a threshold. An estimated threshold value of 5% of the most frequent relevant term has shown to provide good results. Additionally, the discarded terms are kept available for adding them again manually.

term	weight	frequency relevant set	frequency input set
tropical	66,5	1750	4256
hurricane	64,9	69714	93716
storm	63,8	423	830
irene	60,9	11696	47381
surge	60,2	431	1537
evacuated	50,0	165	1759
staying	45,0	264	9879
place	40,0	519	52856

Table 1. Selected top, medium, and low weighted terms from a set of messages during the hurricanes Sandy and Irene. The set of relevant messages was defined by searching for `storm` and `hurricane` and is a subset of the input set. The latter was generated by a geographic filter on the US east coast.

The definition of relevant messages so far only depends on the rough initial query, but it already provides an overview of frequently used terms that the analyst was probably not aware of. Therefore, it is possible to iteratively refine this set with the current result by widening it with additional query terms. Further, the co-occurrence of terms can be exploited to use only special meanings of a word that result in the combined use with another term, e.g. `flu` and `shot`. This results in a weighted keyword list that can be applied to messages by summing the normalized term weights of terms that appear in the list and the weights of normalized co-occurrences. When applying the filter during the monitoring phase, a user-configurable threshold defines a minimum weight that determines whether messages are to be included in the filter result. Here, a threshold of 0 would include every message, while a threshold of 1 would only include messages that consist of terms with the highest weight. The default value is set to 0.5.

3.2 Classifier Creation

ScatterBlogs2 enables analysts to construct filters by providing a training environment for message classifiers. The task of finding terms relevant to an event is replaced with the task of providing suitable training examples. If the analytic focus lies on low-frequency incidents, classification approaches are likely to outperform clustering approaches, especially in scenarios dealing with huge amounts of dynamic data such as microblog messages. On the one hand, data driven methods, such as clustering, typically require a sufficiently large number of messages to identify clusters/outliers, which is likely to delay the detection of low-frequency incidents, rendering them unsuitable for live-monitoring situations. On the other hand, clustering approaches creating such a context must take into account the dynamic nature of microblog systems to stay scalable. Clustering algorithms relying on a predefined number of clusters might not adapt adequately to dynamic developments and clustering methods depending on inter-message similarity are often limited with respect to scalability, due to quadratic runtime complexities or expensive rebalancing of data structures. Classifiers instead, can be specifically trained to detect microblogs of interest by training them a previously known set of relevant messages, making them the candidate of choice for detecting low-frequency events of interest. Nevertheless, clustering can help to provide a suitable context to embed low-frequency findings and support analysts in assessing their findings in relation to the overall situation. In previous works we developed a stream clustering based anomaly detection method, that can still be employed to embed low-frequency analysis within a context of detected high-frequency behavior [25].

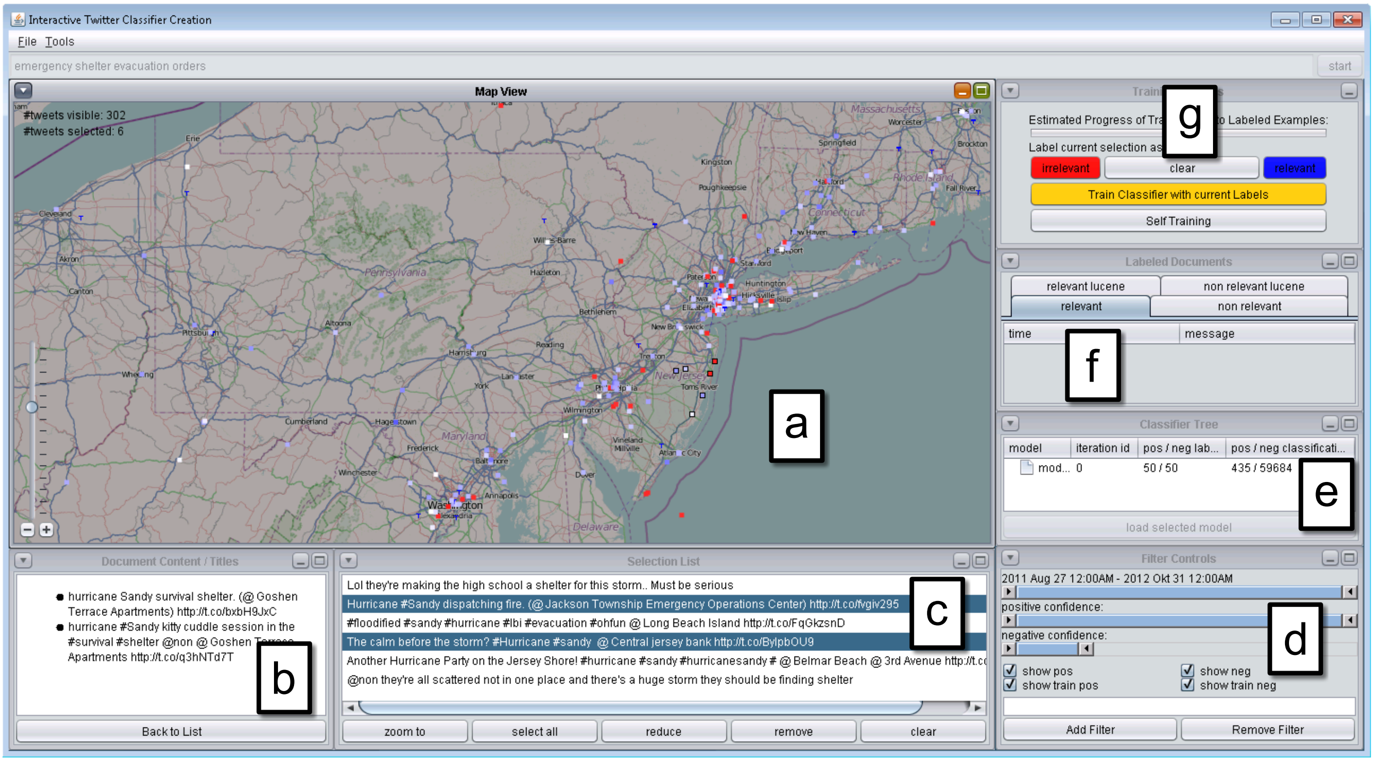

Fig. 2. The classifier creation environment of ScatterBlogs2 with the Map View (a), the Message View (b), the Selection View (c), the Filter Controls (d), the Classifier Tree (e), the Labeled Documents List (f), and the Training Controls (g).

3.2.1 Classification Environment

We use the support vector machine framework [30], which is known to perform well on textual data, and we provide two different kernels that analysts can use for their classifiers. One is a linear kernel for boolean message vectors that offers fast training and decoding times. Due to the noisy nature of microblog messages, term-based classification methods can cause problems if applied to texts containing typos or colloquial language. Therefore, we additionally provide a string kernel [12], which directly compares message strings to assess their similarity. String kernels offer robust classification in the presence of typos and other noise. Their computation, however, is more time consuming leading to longer training and classification times.

The training environment of ScatterBlogs2 is depicted in Fig. 2. It offers visual support to interactively create binary support vector machine classifiers, which can detect microblog messages on specific events from a continuous message stream. To quickly bootstrap an initial classifier, highly ranked messages from the result set of a coarse keyword query can be used as positively labeled messages and some arbitrary posts not returned with the result set as negative examples for training. Afterwards, analysts are supported in exploring, finding, and labeling messages according to whether they are relevant to their information need or not. Rather than letting users blindly label huge numbers of messages, our training environment provides feedback on the classifier's progress after each training iteration. This spares analysts time-consuming and tedious work where success can only be evaluated after enough time has been invested to label sufficiently large training and test sets. During each training iteration, new messages are labeled, and a new, updated classification model is created. Visual feedback on the resulting, intermediate classifier is then provided in terms of its classification decisions on the remaining unlabeled instances. The techniques we employ in this training environment are based on those in [10], with significant changes to adapt it to microblog messages. Some views, namely the Message View (Fig. 2b), the Classifier Tree (Fig. 2e), the Labeled Documents List (Fig. 2f), and the Training Controls (Fig. 2g) have been directly adopted with minor changes from previous work. All adaptations to those views and the newly developed views are described in the following section.

3.2.2 Creating Classifiers Interactively

After a recorded set of messages has been loaded, analysts can start creating a classifier. Having created an initial classifier through the mentioned bootstrapping step, users are supposed to start exploring the Map View (Fig. 2a), in which the messages are depicted as small, colored glyphs at the location they were sent from. Messages classified as relevant are colored in blue, those classified as irrelevant in red. Classification confidence is encoded by brightness with higher brightness meaning lower confidence. The shape of the glyph depicts the state of the message. Rectangles represent unlabeled messages, triangles represent messages that have been labeled during the current training iteration, and T-shaped message glyphs represent training examples labeled during previous iterations.

In the Map View, users can mark and select messages to acquire additional information about them within the brushing and linking based environment. Messages are highlighted when hovered with the mouse or the message lens. The size-adaptable lens displays high-frequency hashtags in the set of hovered messages to explore the popularity of hashtags in different regions. The content of highlighted messages and the time they were published is displayed in a separate view (Fig. 2b). Messages that were selected in the Map View can be labeled either as relevant or irrelevant using the Training Controls (Fig. 2g). A List View (Fig. 2c) displays the texts of all selected messages for closer inspection before labeling them. Using this view, the set of selected messages can be constrained further by removing unwanted messages. With the List View, analysts can, e.g., select specific areas from the map where an interesting event took place, remove all unrelated messages, and label the remaining relevant ones.

The Filter Controls (Fig. 2d) offer further assistance in finding relevant messages. They include a range slider for constraining the publishing date and time of the messages displayed to show only messages published during the known time frame of an event. A replay of the arriving messages can be achieved by constraining the creation time of displayed massages to a small timespan and then moving the slider. Thus messages pop up in the order they were published to trace, for example, the route of a storm. The Map View in combination with the temporal filtering enables analysts to find past events by their spatiotemporal extents. Two additional range sliders let analysts restrict the confidence range of positively and negatively classified messages on the Map View. This makes it possible to use different labeling strategies, e.g., concentrating on low confidence messages, which is known as uncertainty sampling [21], a strategy that works well for SVMs [27]. Furthermore, information on low confidence classifications is useful to judge the quality of the current classifier, and thus helps to assess training progress. The Filter Controls further allow to turn the display of positive, negative, and training examples on and off separately and to filter messages by keywords. The latter is useful to find messages containing a certain hashtag or place name.

To further speed up the classifier creation, the training environment offers self training, a technique using the information from unlabeled examples for classifier training [2]. It refines a classifier over multiple training iterations by automatically labeling examples classified with highest confidence with the label suggested by the classifier. After starting self training via the Training Controls (Fig. 2g) and defining the number of iterations, users can keep track of the training progress visualized on the map and stop the automatic process at any time. While self training can lead to overtraining if the dataset is skewed and repetitive, we found that with supervision by the analysts, who can manually interfere during training iterations by labeling new data, self training leads to a more robust classifier with better generalization.

4 Monitoring Environment and Filter Orchestration

For live monitoring we provide an integrated analysis environment (Fig. 3) that can be used by operators in three activity phases. All three activities heavily rely on the system's support for orchestrating keyword and classifier filters described in Section 3.

(1) In its default operating mode the system serves as a monitoring device in control rooms and similar environments. The operators would usually apply a recall-optimized selection of filters trained to detect messages relevant to their domain and task. When an unusual event occurs, the system guides the operator's attention by providing visual cues for detected messages. (2) This triggers the second activity, in which the operator would use the system to gain an overview of the ongoing situation. The second activity is supported by means for visual document aggregation and interactive spatiotemporal search and manual filter operations. At this point the operator can also apply more specific filters adapted to the ongoing situation and combine them with manual filters that help to reduce noise or widen the detection to more situation-specific topics and areas. (3) When analysts have developed such situation-specific information needs or require a more detailed picture of an event's extent, they can perform an in-depth analysis on available messages. This last activity is supported by a range of visual tools for spatiotemporal exploration and examination. In this phase, the filter orchestration is primarily used for selection management and capturing the analysis' provenance.

In the following Section first introduce the user interface and visual tools of the general monitoring and analysis environment. In Section 4.2 we describe the orchestration process and interaction mechanisms, which are provided through an individual view within the interface.

4.1 Monitoring Environment

The monitoring environment offers a live visualization of the twitter stream, ad-hoc keyword and geotemporal filters, and tools that allow the exploration of the textual content of message sets. The results of the filter orchestration are shown in this interface.

4.1.1 Stream Visualization

In our approach, we put special focus on location-enabled data. Especially for length limited microblog messages, the spatial domain can establish the necessary context to interpret message distributions of topics and terms. The monitoring system is thus designed around a zoomable world-map visualization showing locations of recent messages as scattered points on their respective geolocations. We specifically listen for geolocated data via Twitter's filtered streaming API [28], receiving about 91% of all geolocated messages worldwide and totaling to about 7 million messages per day. Every time a message is written by a Twitter user, a yellow spot appears on the map to indicate new data to the operator. After a short while the yellow spot slowly blends into a persistent small red spot.

4.1.2 Search, Filter, and Selection

Our monitoring environment provides support for keyword and geographic search as well as means for manually customizing spatial and temporal filters. Keyword search can be performed using a text box and keywords can be combined using boolean operators. The user can also look up names of geographic places[2] on which the map should be centered.

If users are only interested in specific geographic areas, they can sketch these areas on the map by polygons to filter out the data from other regions. Similarly, the message volumes of different time periods can be compared by using a resizable time-range slider to filter temporal sections. The slider can then be moved in a continuous fashion to provide an animated overview of the past development. The temporal view of this slider is separated into three detail levels showing histograms of message volumes by day, hour and minute (see Fig. 3f). This allows to maintain orientation while browsing large time spans on a detailed or broad scale alike. More details on this component can be found in [34].

Since the capabilities of detail views are limited, we adhere to a two stage *filter and selection* scheme. Certain views, such as the map, will instantly show the result of textual search and spatiotemporal *filter* operations. If users need to investigate filtered messages in detail, they can set the current filter as *selection*. Views like the content table (Fig. 3), which will allow a more detailed analysis, will then be updated to the current filter set.

4.1.3 Content Exploration and Examination

For the second and third activity of the monitoring and analysis process we provide additional content aggregation approaches supporting operators in getting an overview of available data and explore them on a spatiotemporal basis. These tools have already been featured in previous works for post-analysis scenarios [4, 6, 25] and will therefore be introduced only briefly here.

Contentlens: If analysts want to explore message contents in relevant geographic areas, they can summon multiple lens-like tools (see circle in Fig. 3), that can be dragged over the map and continuously highlight the most prominent keywords currently used in the respective area. In a second mode of operation a spatial version of tf-idf [24] is used to remove terms that are rather common to the area like city names and show only terms of unusual prominence. During monitoring, this technique is valuable for providing details on demand about both low and high frequency events.

LDA Topicview: Once operators have identified an interesting message subset, they can activate an LDA based [3] analysis of prevalent topics for deeper analysis. The detected topic clusters will be shown in a sidebar as a list of small tag clouds representing a bag of words for each isolated LDA-topic (see Fig. 3b). Topics are not extracted from the whole stream, due to the complexity of LDA analysis. Instead, each user selection is analyzed and potential sub-events are presented.

4.2 Orchestration of Filters During Monitoring

Filters are constructed using previously observed data, and even if the combination of several instances of an event type during training helps to generalize the filter constraint to unseen events, the filters still need

[2]This feature is provided through geonames: http://www.geonames.org/

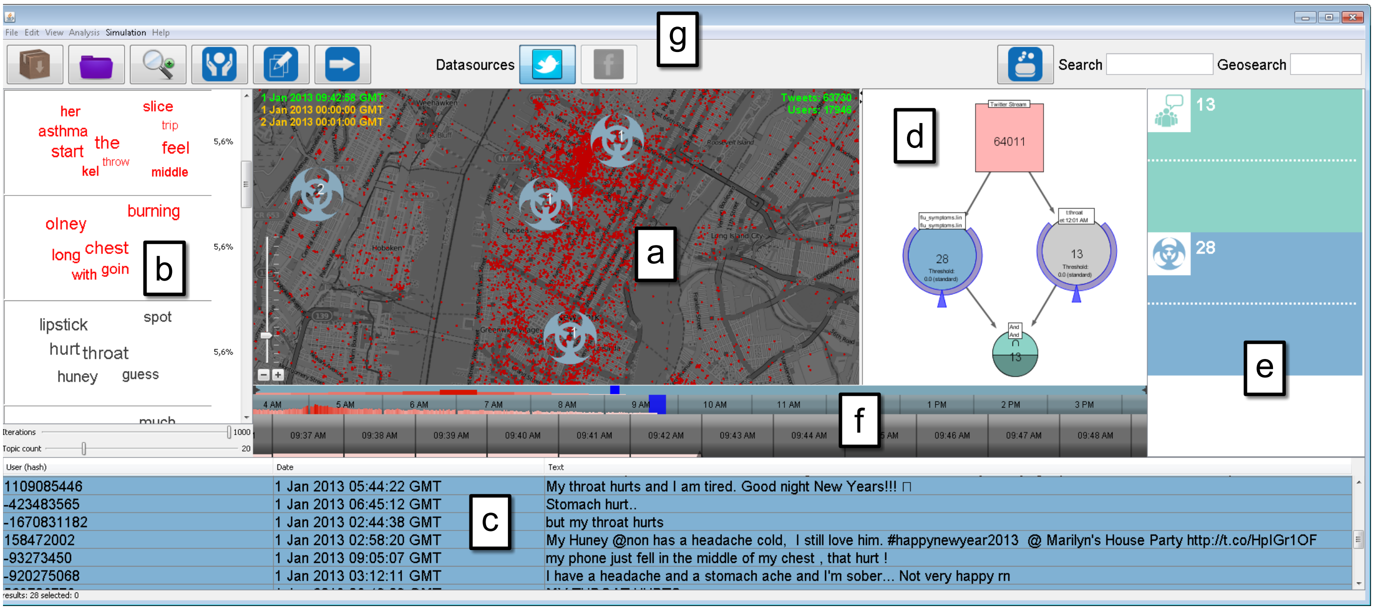

Fig. 3. User Interface of the monitoring environment showing (a) the Map View with aggregated filter detection symbols with the respective number of aggregated messages, (b) the LDA topic view, (c) the table of selected message contents, (d) the filter orchestration view, (e) the class panel view, (f) the smooth scroll time slider, and (g) the control panel allowing to perform textual and geographic search, summon Contentlenses, and starting simulation mode.

to be adapted to the current monitoring situation. This can be exemplified in three ways:

(1) During monitoring sessions in which no message passes a filter, operators should be able to judge whether no event occurred or whether there are relevant messages that are erroneously filtered out. In these situations, they might want to widen the filter's focus and inspect the closest hits to make sure nothing important is missed. In the opposite case, when too many relevant messages are present and pass the filter, the observer might want to sharpen the filter's focus to be able to handle at least the most important ones. (2) When additional details about an emerging event unfold, they can be used to exchange or add more specific filters to handle this type of event adequately. (3) In the case of unforeseen anomalies like trending tags that erroneously trigger important filters, these filters could be combined ad-hoc with preprocessing metadata or keyword filters that handle those messages. This avoids retraining filters to adjust them to the current situation.

We see further benefits in an ad-hoc orchestration of prepared and dynamically defined filters. These are, firstly, a separation of concern and expertise between training the filters and applying them during monitoring. This can lead to better filters that generalize over more events and can improve the ease of learning the user interface . Secondly, it provides the monitoring user with a way to chain filters and split result sets in order to observe different and smaller aspects of an ongoing event. To realize these capabilities and convey the effectiveness of the active filters, the orchestration component must display relevant figures and integrate well with the other components of the system.

Therefore, ScatterBlogs2 has a visual filter orchestration based on a graph editing canvas that is used to model a filter/flow graph. This graph has a single, persistent root node receiving the unfiltered incoming message stream. This node is set apart from the other nodes by its rectangular shape (see Fig. 3d). All other, user-added nodes have an associated filter that reduces the incoming data and outputs the result as new input for other nodes. In this setup, each node can have multiple listening child nodes but only one incoming edge, resulting in a tree structure. In order to merge separate flows of the tree again, there are special combination nodes which implement symmetric, n-ary set operations such as union, intersection, and symmetric differences. To these, the number and order of incoming data streams is irrelevant. This simplifies the interaction because only the adjacency of nodes defines the combination result and not the order of creation, as it would be the case for non-symmetric operations like relative complements.

In total, four basic operations are needed when interacting with the graph editor: creating/deleting nodes, creating edges, changing thresholds, and tagging nodes. Nodes are created or deleted via a context menu that allows to load previously saved filters or to instantiate the available stream combination nodes. Connections between nodes are created by dragging the source node onto the target node. During the drag, valid targets are highlighted in green and the creation of circular dependencies is prevented. If a specific filter supports the adjustment of a threshold value, it can be changed directly at the node using a circular slider in a normalized range $[-1;+1]$. The filter implementation is then responsible for translating this value into a meaningful threshold value that maximizes recall for the value -1 and precision for the value $+1$. For instance, the weighted keyword filter can directly map this value to its internal weight threshold, while the SVM classifier can use it to adjust the bias of the decision border.

The integration of the orchestration component with the rest of the monitoring system is realized by *tagging*. A tag consists of a user-defined name, color, and icon and is assigned to a node via a context menu. If a document passes the filter of a tagged node, it collects the tag and will hence be marked throughout the system by the associated color and icon. The tagging of a nodes content which is defined by all filter criterions along the paths from the root node is the key element of describing and monitoring current events and situations. The Map View displays tagged messages with their associated colored class icon and aggregates similar icons in close vicinity of each other to one single icon, labeled with the count of aggregated messages (see blue icons in Fig. 3a). The link between the tagged messages and the tagging node is established by filling the node's shape with the color of the tag and labeling it with the tag name. Additionally, there is a special `current` tag, which, if assigned, functions as a visibility filter and hides all other messages throughout the monitoring interface. This is helpful as a drill-down operation into a subset of messages for closer inspection.

Fig. 4 depicts a node and its context. The topology of the graph represents the flow of messages and the working sets of each filter node. The node itself shows the details of its configuration: Its type

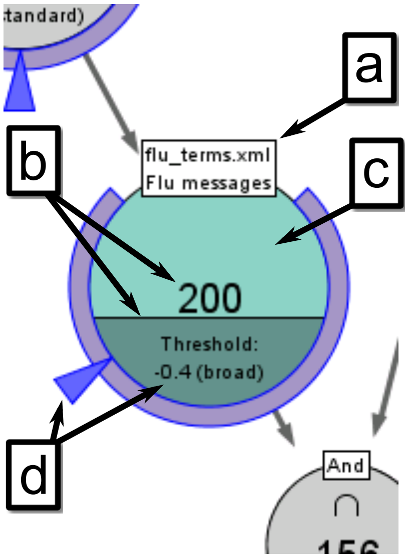

Fig. 4. A node in the orchestration graph with indications of: the node type and tag name (a), the selectivity (b), tag color (c), and threshold settings (d).

and tag name, if assigned, constitute a label for the node (a). The absolute number of messages that passed its filter are shown in the center of the node, while the "selectivity" (percentage of incoming messages that pass it) is denoted by using a darker fill color for the appropriate portion of the node (b). Here, the color corresponds to the tag or is gray otherwise (c). Finally, the current threshold setting is marked with an indication in natural language of what the numeric value roughly means (d).

Implementationwise, the stream enabled filter graph is represented as message sets at each node that contain the messages that passed the filter. Each child node listens to changes in the result set of its parent and reacts to changes by evaluating its own filter on new content in a separate thread. Thereby, set changes propagate through the graph independently and in parallel execution.

5 Case Study

This section demonstrates the feasibility and applicability of our approach with two case studies. The first one depicts the interactive, visual creation of a classifier for detecting messages from users reporting on flu-related symptoms. The second case study describes a fictitious analyst who monitors the abnormal weather situation with heavy rain in Great Britain and Ireland in the fall of 2012. Due to the infrequency of such events, we previously recorded data from the Twitter stream and provide a component to replay them exactly as they were received for demonstration and evaluation purposes.

5.1 US Influenza Epidemic - Filter Creation

During the winter of 2012/2013, the United States suffered from an epidemic influenza outbreak and the state of New York declared a public health emergency in the second week of January. In order to monitor the further development of this epidemic, we used one week of data to train a keyword metric as well as a SVM classifier to identify flu-related messages.

5.1.1 Creating a Flu-Related Keyword Metric

First, we create a corpus of potentially relevant messages by supplying an initial set of keywords: `flu`, `fever`, and `headache`. The system then calculates the term weights, as described in Section 3.1, and presents several ranked lists of keywords and co-occurrences. These can be used to widen the initial query and thus expand the set of relevant messages, or they can be directly included in the definition of a keyword based filter metric. On the first page of terms we find interesting symptom and medication related terms such as `shot`, `outbreak`, `pounding`, `stomach`, `throat`, and `coughing`, that we might have missed otherwise. We collect most of these terms and add them to the filter definition, but restrict `shot` to only count in combination with `flu` and `pounding` with `headache`, which is already suggested by the system in the list of high ranked co-occurrences. Afterwards we widen the query with these selections and iterate the process until we feel confident with the result. In order to evaluate the filter definition, we test it on the collected messages and preview the result. Finally, we store the keyword and co-occurrence list together with their metric weights.

5.1.2 Flu Classifier Training

After constraining the dataset to contain mostly flu-related messages, we further include a random sample of messages that are flu unrelated. While the flu-related messages are potential positive training examples, the unrelated ones will serve as candidates for negative examples during the classifier training. Providing a set of negative training examples that is as representative as possible for the messages contained in the live microblog stream will harden the resulting classifier with respect to those irrelevant messages and increase its accuracy during live monitoring. We bootstrap the classifier as described in Section 3.2 with the previously identified relevant query terms `fever`, `flu`, `outbreak`, `headache`, `throat`, and `coughing`. To inspect the outcome of this automatic step, we set the filter to show only labeled messages and skim through the message contents in the selection list. We find some messages that have been labeled incorrectly by the bootstrapping query and relabel them with their correct class. For the subsequent refinement of the classifier we first take a look at the messages close to the decision boundary by using the filter controls to restrict the displayed messages accordingly. Looking at the messages at the decision boundary allows us to assess the quality of the classifier. Additionally, labeling messages in this region has the highest potential influence on the new classification model. We thus inspect and label messages in this area, and then retrain the classifier. Then, we take a look at the messages classified highly positively and negatively, and find that the current classifier detects them correctly. In order to further improve the classifier we do 100 iteration of self training, and inspect and label instances close to the decision boundary again. After a couple of iteration of alternating labeling at the decision boundary and self training, we find that we are satisfied with the quality of the classifier. The manual part of the training process took about 20 minutes and 374 instances were labeled in 8 iterations.

5.2 Great Britain and Ireland Floodings - Filter Usage

In 2012, Great Britain and Ireland were affected by a series of severe weather events that caused heavy rain and floodings. This resulted in numerous flood alerts, evacuations, road blocking landslides, a derailed train and even fatalities throughout the British Isles. The total insurance losses through flooding for 2012 have been estimated at £1.33 billion GBP [32].

In order to evaluate the applicability of our real-time monitoring environment based on data generated during these events, we collected all georefereced Twitter messages from six days in June where floodings have been reported and use this set to train a classifier to identify flood related messages. We then used this filter in combination with previously created, general purpose classifiers and keyword based metrics to monitor selected days of the year. Based on a replay of Twitter data collected during November 26 — a day on which severe weather conditions occurred — the following paragraphs describe in chronological order how an operator could have monitored and analyzed flood related messages during that day.

Preparation In the beginning of the monitoring phase the operator summons an initial set of emergency related filters trained to detect severe weather effects, fires, damages, etc. To improve the effectiveness of the default filters the operator combines them with filters that will remove spam and news media, or other messages containing second hand information (e.g. retweets).

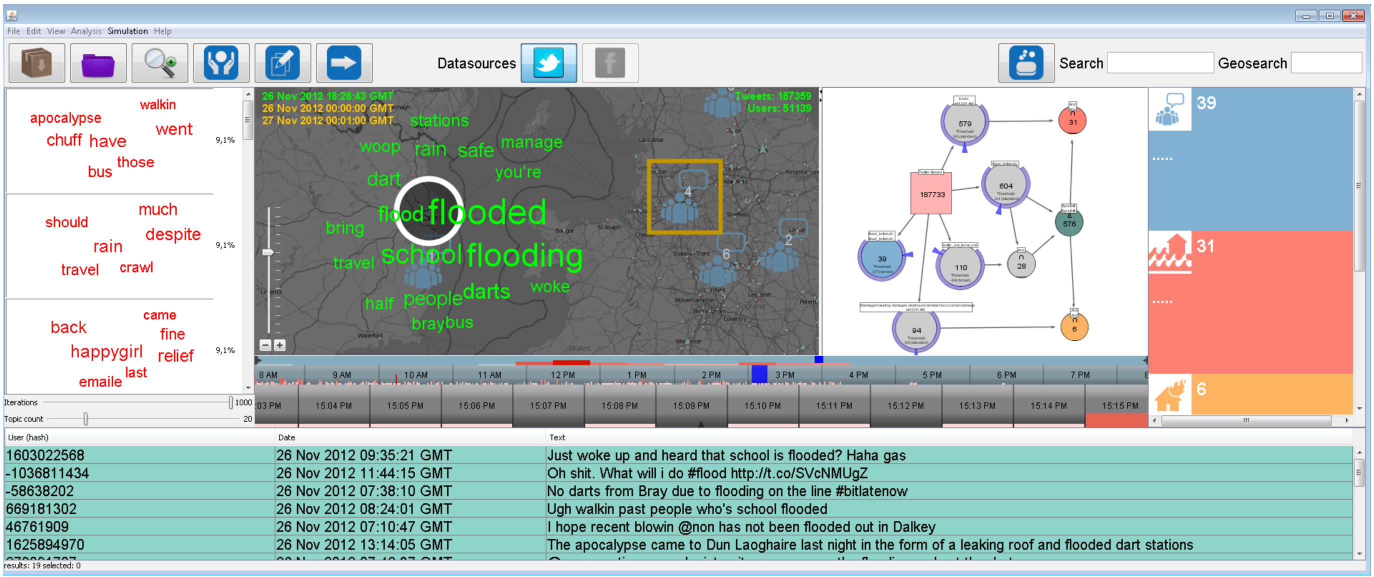

Fig. 5. Application of the monitoring environment during the Great Britain and Ireland Floods. The picture shows a more sophisticated filter graph that the operator has already constructed to find high-profile and low-profile flood related messages and separate them from spam and traffic information messages. In the map view we can observe the application of the Contentlens to the flood related filter context, indicating that schools and DART stations were flooded in Dublin.

01:16 GMT - At this time, the flood classifier detects the first messages related to the flood. It seems, that the government-issued flood warnings were successfully disseminated and are discussed by the public (`Flood warning has been issued for #caversham!`). Increasing the filters threshold omits mere repetitions of warnings and reveals a first indication of actual weather impact appearing at Rathdrum near Dublin (`Our yard is flooded!! Go away Rain!!!!!`). The operator continues with two instances of the classifier, one with a strict threshold for high profile messages and one with a medium threshold for monitoring the overall trend.

06:16 GMT - According to a local resident, river banks broke on the River Swake in Richmond. This is the second message detected by the high threshold classifier. Also, more and more messages of a traffic information service begin to appear within the detected messages. As such messages will mostly convey information already known to the operators they can hide them ad-hoc with a keyword filter on the words used by the service and a symmetric difference between the keyword and the flood classifier filter.

08:44 GMT - The situation begins to unfold as more flood indications start to appear all over the map. The aggregated classifier icons provide an impression of the distribution, but their increasing number hinders the further examination of events. Therefore, the operator creates more specific categories, such as road blockage related messages, by combining the classifiers with manual keyword and region filters.

10:21 GMT - Although unrelated to the flood, the default emergency filters instantiated during preparation show that a fire broke out in Oldbury near Birmingham caused by an explosion at a distillery. Several eyewitnesses talk about the incident and report on its severeness (`Saw the explosion at the oldbury fire as I was driving past on the motorway... .`) which results in a clear peak in the area compared to other parts of the country.

11:07 GMT - The map overview and road blockage classifier icons provide the operator with a good indication that traffic is hindered in southern and middle parts of the UK. Inspecting some of their messages by selecting an icon highlights reports on flooded roads (`Can't believe I drove down a flooded road where I couldn't see the sides or the end. Was like being in a boat.`).

15:41 GMT - Since the operator is already confronted with a very high quantity of flood related messages (about 579 for the default threshold classifier), it is now a good idea to get a general overview of what the people are concerned about in different parts of the country. The analyst thus selects the flood classifier and sets it as the current filter. Based on this filter context, it is now possible to apply visual aggregation tools like the LDA Topic View and the Contentlens in order to find topics connected to the flood. The exploration of the map quickly shows that in Dublin `school` and `dart` are prominent keywords among the flood related messages (see words around the circle in Fig. 5). By investigating these messages, the operator can quickly understand that the Sandycove DART (Dublin Area Rapid Transport) Station is flooded and that schools in the areas have been closed in the morning due to the flooding. Using similar means it is possible to see, that several people complain about delayed or canceled trains because of the weather conditions in London and that the region of Worcester was severely affected by the flood.

As the system helps the operator to always keep an overall picture of the ongoing events, situational awareness is ensured at all times. Although the operator was able to detect several smaller incidents and flood damages that affected people, it was also possible to recognize that the general situation stayed under control.

6 Expert Feedback and Discussion

In order to get an idea on the suitability of the visual monitoring interface, we designed a questionnaire regarding aspects of usability and task appropriateness. The feedback on this questionnaire is limited to the monitoring stage of our approach. The visual means of creating classifiers for document retrieval has already been generally evaluated in [10] on considerably larger texts than microblog messages. There exist no evaluation results regarding the specific adaptations that have been made to microblog data classification or to the SVM employing the string kernel and the statistical keyword widening approach. Due to the complexity and expensiveness of implementing a complete test setup with analysts responsible for classifier creation, gold standard evaluation and practical application by operators during the monitoring phase, we plan to perform a thorough evaluation in such a manner in future work.

6.1 Expert Feedback on Microblog Monitoring

We provided the questionnaire to an expert from the German Federal Office of Civil Protection and Disaster Assistance (BBK) as well as to a usability specialist from Siemens AG, after we showed and explained them the monitoring interface in a twenty minute presentation and a subsequent Q&A round. Both institutions and our research group are part of the German BMBF-funded research project Visual Analytics for Security Applications (VASA).The questionnaire items were formulated as statements where the experts could signalize their strength of agreement/disagreement. The following paragraphs summarize their judgment.

Both experts agreed on the importance of having a real-time system for monitoring tasks during disastrous events and for creating situational awareness. They also agreed on the importance of temporal, spatial, and keyword filtering, and the combination of these methods. Their opinion on how long information should be kept by the system differed. While the usability expert strongly agreed that information should be removed from the system automatically, the disaster management specialist was undecided. They expressed opposite opinions regarding the dependency of message removal according to filter type. The expert for disaster management did not want to give any statement on the usability of the visual perspectives before she could test the system herself extensively. However, the usability expert remarked that all components complement each other very well, but she criticized the organization of filters in a graph layout.

When asking for assessment of the system's suitability for monitoring events as part of disaster management, we received rather homogeneous feedback again, with one exception. While the expert on disaster management felt confident to use the approach in addition to other tasks, the usability expert disagreed in that point. Both agreed that a good overview of the incoming information is given, situational awareness is provided, and that the system would be a useful enrichment for disaster management. Finally, the expert from the Federal Office of Civil Protection and Disaster Assistance mentioned that she would like to employ the system for disaster management tasks.

6.2 Filter Creation

The motivation for using filter methods beyond plain keyword lists and metadata restrictions is generalization and their customizability by adapting thresholds. From our own experience the performance of such filter methods with respect to perceived accuracyfor detecting certain events and message types is rather different. For some information needs, it is possible to create good classifiers and statistically motivated keyword lists, while in other situations we were not able to achieve acceptable performance. The same observation applies to the application of well established methods such as spatiotemporal restriction and direct keyword filtering. All of the mentioned techniques can work fine when used on their own, but typically a combination of them is beneficial. In general most monitoring tasks aim at high recall, meaning that it is important not to miss an important message. Achieving this, however, often requires a trade-off regarding precision. As a consequence we provide the capability to train, combine, and configure classifiers and filters to let analysts decide on this trade-off based on domain knowledge and situation specific demands.

Nevertheless, the creation of a good filter combination requires some expertise and can lead to unintended effects if not done properly. Care should be taken, for example, if keyword filters are applied before machine learning based methods, since these can cause a distinct cut off ruling out the propagation of vast numbers of messages. However, in case of very broad coverage such a strategy can be very helpful, if inadequate results are achieved with classification alone. Building good filter sequences can therefore be seen as a creative act, which requires testing different filter combinations. It is therefore important to provide our orchestration method that enables analysts to create and test filter combinations on their own.

6.3 Scalability and Performance

We specifically focused on scalability aspects during the development of the described approach. These aspects include scalability of filter creation, scalability of filter application and combination, as well as real-time monitoring. The separation of filter creation and application allows the involvement of multiple users specialized in different domains and tasks, such as classifier training or monitoring. Once created, generic filters can be used in an arbitrary number of monitoring sessions and scenarios.

In order to make filter processing and orchestration scalable, all stages of the user-defined filter-pipeline are parallelized. Here, the benefits vary depending on the complexity of the filters to be evaluated. Support Vector Machines with a linear kernel can be evaluated very fast on short documents such as Twitter messages, limiting the benefit of parallelization due to the management overhead.

The most critical points in the pipeline are those filters working directly on all incoming messages of the graph root, because subsequent nodes will only receive input that is usually already strongly reduced. Table 2 contains performance evaluations of single filter types as well as a whole graph structure. The graph is composed by three linear SVM classifiers and one keyword metric filter, which directly work on the whole input data. Their output is joined by an OR/Union node and ends in a geographic filter. The performance was measured on 3.9 million messages using a machine with forty physical cores. The geographic filter node was reached by 31 thousand messages.

Filter type	Time/Message, 1 Thread	Time/Message, 40 Threads
Linear Kernel SVM	$19\mu s$	$6\mu s$
String Kernel SVM	400ms	87ms
Keyword Metric	$20\mu s$	$6\mu s$
Whole Graph	-	$73\mu s$

Table 2. Average time needed for evaluating a filter on one microblog message when evaluating them sequentially or in parallel. The *Whole Graph* is composed of multiple filters and due to the inherent parallelism of the graph structure, no value is given for one thread.

It can be seen from these figures, that the employed techniques are fast enough to have several filter instances running on the global stream of georeferenced Twitter messages on a normal computer. The only exception is string kernel based SVM classifier, which should only be used on already reduced streams. This can usually be achieved, e.g., by constraining the incoming stream to the location of interest.

7 Conclusion

We presented an integrated approach starting from interactive visual exploration of past situations, over creating statistical filter methods efficiently and interactively, to their orchestration during monitoring and microblog analysis. During the training phase analysts are supported by highly interactive means to exploit recorded data from past events in order to design classifiers based on the domain and task specific needs of the operators. Our monitoring and analysis environment subsequently provides sophisticated means to orchestrate, combine and configure the trained classifiers together with ad-hoc filters adapted to the ongoing situation. The applicability of our approach for gaining situational awareness could be shown based on Twitter messages recorded during critical events.

The varying quality of results we achieved with different filter methods when used on their own, and the improvements we gained by combining them indicates that this is a promising direction of tackling the challenge of scalable real-time microblog analysis. Future work is needed to find ways that support analysts in developing strategies to create beneficial filter combinations more quickly or even semi-automatically. We look forward to let analysts test the proposed methods in real monitoring applications in the future.

Acknowledgments

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) in context of the VASA project, by the German Science Foundation (DFG) as part of the priority program 'Scalable Visual Analytics', and by the cooperative graduate program 'Digital Media' of the University of Stuttgart, the University of Tübingen, and the Stuttgart Media University (HdM).

REFERENCES

[1] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao. Twitcident: fighting fire with information from social web streams. In *Proc. 21st Intl. Conf. Companion on World Wide Web*, WWW '12 Companion, pages 305–308, New York, NY, USA, 2012. ACM.

[2] S. Abney. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 1st edition, 2007.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Machine Learning Research*, 3:993–1022, 2003.

[4] H. Bosch, D. Thom, M. Worner, S. Koch, E. Puttmann, D. Jackle, and T. Ertl. Scatterblogs: Geo-spatial document analysis. In *IEEE Conf. Visual Analytics Science and Technology (VAST)*, pages 309–310. IEEE Computer Society, 2011.

[5] 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.

[6] 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 *IEEE Conf. Visual Analytics in Science and Technology (VAST)*, pages 143–152. IEEE Computer Society, 2012.

[7] M. Dörk, S. Carpendale, C. Collins, and C. Williamson. Visgets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1205–1212, 2008.

[8] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *IEEE Conf. Visual Analytics in Science and Technology (VAST)*, pages 93–102. IEEE Computer Society, 2012.

[9] N. Elmqvist, J. Stasko, and P. Tsigas. DataMeadow: A Visual Canvas for Analysis of Large-Scale Multivariate Data. *Information Visualization*, 7(1):18–33, 2008.

[10] F. Heimerl, S. Koch, H. Bosch, and T. Ertl. Visual classifier training for text document retrieval. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2839–2848, 2012.

[11] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann. Inter-Active Learning of Ad-Hoc Classifiers for Video Visual Analytics. In *Proc. Conf. Visual Analytics Science and Technology (VAST'12)*. IEEE Computer Society, 2012.

[12] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. J. C. H. Watkins. Text classification using string kernels. *J. Machine Learning Research*, 2:419–444, 2002.

[13] A. M. MacEachren, A. R. Jaiswal, A. C. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. Blanford. SensePlace2: GeoTwitter analytics support for situational awareness. In *IEEE Conf. Visual Analytics Science and Technology (VAST)*, pages 181–190. IEEE Computer Society, 2011.

[14] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proc. SIGCHI Conf. Human Factors in Computing Systems*, CHI '11, pages 227–236, New York, NY, USA, 2011. ACM.

[15] J. Moehrmann and G. Heidemann. Efficient annotation of image data sets for computer vision applications. In *Proc. 1st Intl. WS Visual Interfaces for Ground Truth Collection in Computer Vision Applications*, VIGTA'12, pages 2:1–2:6, 2012.

[16] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. Intl. Conf. Intelligence Analysis*, pages 2–4, 2005.

[17] J. Roberts. State of the Art: Coordinated Multiple Views in Exploratory Visualization. In *Intl Conf. Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, pages 61–71, july 2007.

[18] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. of the 19th Intl. Conf. World wide web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[19] C. Seifert, V. Sabol, and M. Granitzer. Classifier hypothesis generation using visual analysis methods. In *NDT (1)*, pages 98–111, 2010.

[20] B. Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, EMNLP'11, pages 1467–1478, 2011.

[21] B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[22] B. Shneiderman. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6):70–77, Nov. 1994.

[23] J. T. Stasko, C. Görg, and Z. Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008.

[24] D. Thom, H. Bosch, and T. Ertl. Inverse document density: A smooth measure for location-dependent term irregularities. In *Proc. 24th Intl. Conf. Computational Linguistics*, COLING, 2012.

[25] D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *IEEE Pacific Visualization Symp. (PacificVis)*, pages 41–48. IEEE Computer Society, 2012.

[26] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.

[27] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *J. Machine Learning Research*, volume 2, pages 45–66, 2001.

[28] Twitter. The Streaming APIs. https://dev.twitter.com/docs/streaming-apis accessed on 31-March-2013.

[29] J. van Wijk. The value of visualization. In *IEEE Visualization*, VIS '05, pages 79–86, 2005.

[30] V. Vapnik. *Statistical learning theory*. Wiley, 1998.

[31] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: Finding topic-sensitive influential twitterers. In *Proc. 3rd ACM Intl. Conf. Web Search and Data Mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM.

[32] Wikipedia. 2012 Great Britain and Ireland floods — Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/2012_Great_Britain_and_Ireland_floods?oldid=546752704 accessed on 31-March-2013.

[33] P. C. Wong, E. G. Hetzler, C. Posse, M. A. Whiting, S. Havre, N. Cramer, A. R. Shah, M. Singhal, A. Turner, and J. Thomas. In-spire infovis 2004 contest entry. In *IEEE Symp. Information Visualization*, INFOVIS, 2004.

[34] M. Wörner and T. Ertl. Multi-layer distorted 1d navigation. In *Proc. Intl Conf. Information Visualization Theory and Applications*, volume 2011 of *IVAPP*, pages 198–203. SciTePress, 2011.

[35] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Proc. 33rd European Conf. Advances in Information Retrieval*, ECIR'11, pages 338–349. Springer-Verlag, Berlin, Heidelberg, 2011.