Online indexing and clustering of social media data for emergency management

Daniela Pohl a,⁎, Abdelhamid Bouchachia b, Hermann Hellwagner a

a Institute of Information Technology, Alpen-Adria-Universität Klagenfurt, Universitätsstr. 65–67, Klagenfurt, Austria
b Smart Technology Research Center, Bournemouth University, Poole House Talbot Campus, Fern Barrow Poole, BH12 5BB Bournemouth, UK

A R T I C L E   I N F O

Article history:
Received 28 November 2013
Received in revised form 13 January 2015
Accepted 26 January 2015

Keywords:
Event detection
Information filtering
Online indexing
Online clustering
Emergency management

A B S T R A C T

Social media becomes a vital part in our daily communication practice, creating a huge amount of data and covering different real-world situations. Currently, there is a tendency in making use of social media during emergency management and response. Most of this effort is performed by a huge number of volunteers browsing through social media data and preparing maps that can be used by professional first responders. Automatic analysis approaches are needed to directly support the response teams in monitoring and also understanding the evolution of facts in social media during an emergency situation. In this paper, we investigate the problem of real-time sub-events identification in social media data (i.e., Twitter, Flickr and YouTube) during emergencies. A processing framework is presented serving to generate situational reports/summaries from social media data. This framework relies in particular on online indexing and online clustering of media data streams. Online indexing aims at tracking the relevant vocabulary to capture the evolution of sub-events over time. Online clustering, on the other hand, is used to detect and update the set of sub-events using the indices built during online indexing. To evaluate the framework, social media data related to Hurricane Sandy 2012 was collected and used in a series of experiments. In particular some online indexing methods have been tested against a proposed method to show their suitability. Moreover, the quality of online clustering has been studied using standard clustering indices. Overall the framework provides a great opportunity for supporting emergency responders as demonstrated in real-world emergency exercises.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Access to information is fundamental during emergency management in order to deal efficiently with different sorts of incidents (e.g., traffic accidents, hurricanes, earthquakes, terror attacks). Collecting this information is not always an easy task, especially when relief units are not immediately on-site, e.g., due to the distance or street damages. Social media (e.g., Twitter) offers a new opportunity for supporting emergency management by enabling collection of data.

Studies [1,2] show the potential of social media in different emergency situations. People report on any kind of emergency situation they witness. Therefore, social media has become an important instrument to exchange information, thus providing additional perspectives on emergency situations [3].

However, intelligent analysis methods are needed to relieve emergency responders from a cumbersome manual browsing task through this data, which is potentially noisy. Methods should be able to summarize the ongoing situation and provide an overview of the emergency situation at hand. In this paper, we focus on the detection of sub-events, i.e., specific crisis-related hotspots (e.g., flooding in a specific district of a city, power outage in another district) that emergency personnel should be aware of when organizing their intervention.

In our early work [4], we examined clustering algorithms for their suitability to detect sub-events from social media. We used Flickr and YouTube data for aftermath analysis of the crisis situation. In particular our investigations relied on offline clustering which is inappropriate for real-time analysis during the emergency situation.

We introduced an online sub-event detection mechanism [5] which combines real-time clustering and online indexing (i.e., weighting and selection of indexing terms). The sub-events (clusters) are detected and tracked as new items from social media users become available. In [5], the mechanism is used to analyze data from the Hurricane Sandy 2012 in the form of batches. It handles data

Please cite this article as: D. Pohl, et al., Online indexing and clustering of social media data for emergency management, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.01.084
collections from Twitter, Flickr and YouTube. We extract terms as features from the textual metadata of the incoming items. We do not process videos from YouTube and images from Flickr and do not analyze their contents, we rather extract their textual metadata (title, description, and tags) to be used along with tweets. Initial experiments on this data show the suitability for detecting topics related to the crisis at hand.

We integrated our online detection mechanism in a media exploration framework. For evaluation of the online processing method, we implement similar indexing methods and compare them with our indexing approach. Hence, the focus of this paper is on the examination of the indexing methods. In doing so, we adapted the online clustering algorithm described in [6] to meet the context of our present application. The experimental setting and the results regarding the different methods are described. They emphasize the suitability of our idea of online indexing for processing social media data.

The structure of the paper is as follows. Section 2 discusses the related work. Section 3 addresses the terminology, i.e., difference between events and sub-events, and highlights it in the context of topic detection and tracking. Section 4 introduces our suggested “Multimedia Exploration Framework”. Section 5 outlines the online sub-event detection, especially the interrelationship between online indexing and clustering algorithms. Section 6 describes the details of the online indexing (i.e., implementations and our learning and forgetting model). Section 7 depicts the used online clustering algorithm in this context. In Section 8, the experimental setting and the results are presented. Section 9 concludes the paper.

2. Related work

The present work is related to “topic detection and tracking” in the area of social media and to “indexing and feature selection” methods.

2.1. Topic detection and tracking

In fact, Twitter is very popular in social media analysis and detection. For example, Gao et al. [7] present an approach that colors geographical regions (social pictures) based on their importance for the topic of interests given by the messages related to these areas. The aggregation and coloring are based on a predefined algebra. The algebra also allows the combination of different social pictures (i.e., with multimedia processing like convolution or segmentation).

Lampos and Cristianini [8] identify important keywords from auxiliary sources, e.g., Wikipedia. These keywords are searched in tweets and scored according to the amount of keywords in the tweet (e.g., to identify the daily fl-rate based on incoming tweets) [9].

Krstajic et al. [10] show an event detection mechanism based on different scores that are calculated and combined by the preferences of the user. First, terms are extracted and combined to episodes (i.e., sets of tweets). After a predefined number of tweets shown to the system, the scores are calculated. If the combined score reaches a threshold, the episode is shown to the user as a new event.

Chakrabarti et al. [11] describe a detection mechanism based on initially learned terms and their importance for a specific event (e.g., football game). In contrast, Shen et al. [12] base their detection on general concepts (e.g., name of companies or persons). General concepts are aggregated together based on their contextual and lexical similarity. Tweets depending on the resulting bag-of-words clusters are analyzed via spike detection and shown to the user. Marcus et al. [13] summarize or identify events based on the peak detection mechanism (covering a one-minute time window). Klein et al. [14] analyze tweets in real time for emergency management. They introduce a graph analysis approach. It allows them to identify leading writing users as the origin of the information spreading. Cataldi et al. [15] describe in their work also a topic detection mechanism for Twitter considering the relation between users, i.e., followers. However, in emergencies people can write about the same events although there is no relation given between them.

Allan et al. [16] describe an approach for detecting and tracking specific events. Nallapati et al. [17] use agglomerative clustering to identify events in a static manner. Osborne et al. [18] describe an online story detection mechanism based on Twitter which uses Wikipedia to verify the identified stories. The framework described by O’Connor et al. [19] analyzes previously fetched tweets to identify and summarize topics. Starbird [20] introduces Tweak-the-Tweet, which defines and uses a predefined grammar for tweets to analyze them accordingly. Twitchicient, by Abel et al. [21], is based on predefined keywords or manually inserted rules. CrisisTracker by Rogstadius et al. [22] (based on [13]) represents a crowdsourcing tool to support volunteers in processing of messages coming from the public during a crisis. It uses an initial term frequency-inverse document frequency (tf-idf) model based on a sample set of tweets.

Most of the approaches use additional or auxiliary material for detection, e.g., Wikipedia, previously processed training sets, or are based on a static analysis. Most of them (e.g., Wikipedia entries or a training set) are often not available during emergencies, especially in fast evolving scenarios.

2.2. Topic detection and tracking based on visual items

In addition to microblogs and text messages, visual items are important in the context of crisis management. Visual items (e.g., pictures and videos) give additional insights into the incident. For example, Chen and Roy [23] perform event detection based on tags annotating Flickr images. The approach allows them to identify periodic and non-periodic events. The tags are examined based on their temporal and spatial distribution and aggregated if they are similar (i.e., representing the same event). The approach allows them also to uncover the time and location of an event. In [24] an approach is proposed to identify disaster events from Flickr. It identifies bursty tags in a predefined time interval and fetches images related to a predefined number of tags. Rattenbury et al. [25] make also use of tags to identify events from Flickr. The identification process is based on a clustering algorithm that takes into account the distribution of tags over time. It is based on specific intra and inter-cluster relationship metrics to identify event-related clusters.

Another approach from Liu et al. [26] identifies events in Flickr images based on the number of items per day coming from unique users. If the number of the incoming items is above the median, a new event is declared. Petkos et al. [27] identify events from Flickr images/items based on Support Vector Machines. Support Vector Machines are proposed to decide/classify if two items belong to the same event. A graph representation is created, where nodes represent items and edges indicate if two items belong to the same event based on the decision of the Support Vector Machines. Community detection algorithms are applied to assign items to events. Rabbath et al. [28] investigate event detection from Facebook by locating photos of the same event shared by friends.

These studies use tags associated with images and were sometimes combined with visual features extracted from the images/videos (e.g., [28,27]) to detect events from social media. In a step forward, we rather use microblog texts in addition to textual annotations of the images and videos.
2.3. Event detection and clustering in image/video data

Analysis approaches used for event detection in image/video streams are specific to the visual nature of the data. There, the visual content of images is used which differs from that used in the social media context of this work.

For example, Zha et al. [29] deal with the detection of events from video streams, e.g., received from surveillance cameras. The authors examine surveillance videos, e.g., from a building in a university. In this context, events are specific group activities, like queuing, discussing, joining, leaving, etc. The approach is based on a new feature called vigilant area (VA) and a graphical model [29]. The VA feature is independent from object tracking conditions and is based on space and time information considering also shape and vigilance [29]. The approach introduced by Ke et al. [30] allows them to identify specific events from videos (i.e., waving with arms or picking up something from the ground). It can detect those events from videos with crowded and cluttered background. The approach covers a shape-based matching algorithm operating in time and space dimensions of the examined video [30]. Schüldt et al. [31] discuss an approach to identify specific actions of people in videos (e.g., running, walking, and boxing) using several spatial-temporal features. Support Vector Machines are trained with these features to recognize the actions within new videos. In addition, a survey on activity detection summarizing different features and recognition approaches can be found in Aggarwal and Ryyo [32].

Image clustering approaches are used for several purposes in research. For example, Zha et al. [33] introduce an image search engine which combines text and content-based image analysis to find images of interest (e.g., apple). First, the system identifies additional related keywords (e.g., fruit and mobile phone) from an existing platform (i.e., Flickr) to refine the user search query. Second, for each identified keyword, the system suggests exemplified images to improve the results of the query. The system combines the Affinity Propagation and the k-mean clustering approach to return images that match to the meaning of the keyword [33]. Chen et al. [34] use image clustering to structure the result list of image search. The idea is to cluster semantically related images by using the k Nearest-Neighbor approach. Hence, a list of image clusters is shown to the user as result. Papagiannopoulou and Mezaris [35] apply clustering for image collection summarization. The authors test different visual features (e.g., SIFT) and clustering algorithms (e.g., k-means algorithm). In addition, Jaffe et al. [36] describe a summarization approach of large image collections based on hierarchical clustering. The authors use different features for clustering (e.g., geo-information and textual information).

Event detection is popular in various applications and uses different techniques such as clustering for surveillance monitoring and repository summarization.

2.4. Indexing, weighting and feature selection

Processing of natural language (e.g., tweets) results in a huge number of indexing terms/features. Feature selection mechanisms are used to deal with such a big feature space. In clustering, feature selection is based on the inherent structure of the data (e.g., frequency metrics) as there are no labels. Witten et al. [37] describe in the context of the WEKA framework [38] several feature selection mechanisms based also on the intrinsic data characteristics. Yang and Pedersen [39] analyze different metrics for clustering (e.g., document frequency, mutual information, and information gain). Liu et al. [40] describe feature selection for clustering based on an Expectation Maximization (EM) algorithm. Additionally, the work by Liu and Yu [41] describes such feature selection mechanisms for clustering. Also, the works in [42] and [43] cover feature selection in the clustering context. Ahmed and Xing [44] extend the Hierarchical Dirichlet Process for describing the evolution of topics and their related words. In addition, Alelyani et al. [45] give an overview on feature selection for clustering.

Brants et al. [46] describe the creation of an index via incremental term frequency-inverse document frequency (tf-idf). Also Marcus et al. [13] make use of the incremental tf-idf. Khy et al. [47] use tf-idf in the context of clustering aging documents. Lee et al. [48] include a statistical function (i.e., skewness) to identify “bursty keywords” in the context of tweets. Also, Lampos et al. [8] make use of such important and frequent keywords for tweeting analysis. Shen et al. [12] identify important terms (e.g., representing specific concepts like names of persons and organizations) for clustering information. Devaney and Ram [49] show a clustering system (i.e., COBWEB) that guides the feature selection process. Singhal et al. [50] show an approach for pivoted tf-idf in the context of profile learning.

Also, a weighting mechanism can be used to perform feature selection or indexing, which allows the modeling of changing topics in streaming data. These weighting mechanisms are mainly coming from the classification area (i.e., labeled data forms the base for the selection process). Bouchachia and Mittermeir [51] describe a feature weighting mechanism for fuzzy classification. Wu et al. [52] show a feature selection method for streaming applications. Relevant features are selected using conditional independency. Guan and Li [53] show incremental features based on changing the topology of the underlying neural network. Rückstieß et al. [54] show an online feature selection in the context of classification. Also, Katakis et al. [55] describe an approach for dealing with concept drift in text-based data streams. In the classification context, they use labeled data and class-to-term statistics. However, this information is not known in emergency management applications since the exact classes (i.e., incidents) are usually not known in advance.

In this contribution, we focus on clustering methods, which do not need labeled data or additional effort before starting the analysis (e.g., training period). Additionally, we consider a weighting mechanism for a smooth feature selection and reduction. Our goal is to identify important sub-events, i.e., hotspots of the crisis, in an online manner and group new incoming data according to these sub-events for emergency management support. In this contribution, we compare our weighting approach with two others adapted for our needs (see Sections 7.1 and 7.2).

3. Terminology

An event (i.e., the crisis itself) is described by its time and location (e.g., Hurricane Sandy in the USA in 2012). It is defined as follows:

Definition 1. An event is a real-world crisis related to a specific time and location.

In general, events can be seen as a composition of smaller parts, called sub-events. Sub-events correspond to different incidents of a crisis, e.g., flooding, damages, and power outage. Sub-events originate in the context of the parent event showing specific hotspots or incidents on a smaller scale, both in space and in time, e.g., flooding in Lower Manhattan NY. A sub-event summarizes postings covering similar incidents several people are writing about. A sub-event is defined as follows:

Definition 2. A sub-event is a specific incident that originates in the context of an event (the crisis).

A sub-event can be identified by incoming information describing the same specific incident, e.g., reports, tweets, pictures, and videos. Sub-event detection can indeed be seen as event detection, as discussed in Section 2. However, in our work we prefer to define...
“event” as the major happening (e.g., “Hurricane Sandy”) which is known to us. Sub-events are those hotspots that collectively form the event. In other words, a sub-event is something that occurs in a certain place, as part of an event, during a particular interval of time. Therefore, we do know the event, but we do not know those hotspots, hence it is important to detect sub-events automatically. Our aim is to uncover sub-events that are triggered by a parent event using social media. The idea is similar to “topic detection and tracking” (TDT) [17], where a topic consists of different events (sub-topics) that are triggered by the topic. We use “events” and “sub-events” as a more specific emergency-related terminology.

We also have a specific spatial-temporal focus on the data itself. A data item used in our identification process consists of geo-information (i.e., longitude and latitude) and textual data of the incoming items (e.g., the tweet itself or metadata annotations related to a picture or video). Like with TDT, we have a fast evolving situation depending on the nature of the crisis. In addition, it is very difficult to label data items in real time due to the nature of an emergency.

4. Multimedia exploration framework

In [4] we suggested a Multimedia Exploration Framework (MEF) for social media analysis in emergency management. We extended the framework from [4] with additional functionalities (see Fig. 1) including online analysis, geo-tagging, and an interactive visualization of the results.

The current version of the MEF now enables both online analysis and offline analysis of social media data. The streaming interface allows data to be fetched in real time. Additionally, we included a geo-tagging mechanism which can be used to automatically tag incoming information. This is performed via named entity recognition [56] related to a location. We use this mechanism to enrich the Twitter data with pictures and videos gained from Flickr and YouTube as those items are very sparsely annotated with geo-tags but valuable for emergency management [57]. This allows us to combine input from different social media platforms, e.g., pure textual information but also textual annotation of visual items. Our system has the possibility to include live data from the incident site (e.g., collected directly from first responders). In the future, we plan to involve additional sources, e.g., data gained from news repositories.

Sub-event detection is based on online feature selection and online clustering as described in the subsequent sections. Section 5 shows an overview of the combination of online indexing and online clustering. The identification of sub-events is based on predefined time periods. The identified sub-events are labeled using the most-relevant terms in the corresponding cluster centers (summarization and labeling stage) and the results are stored in a database for later usage, e.g., browsing.

The sub-events are plotted into a map and visualized to the user in a web-based interface (see Fig. 6). Via a timeline it is possible to browse also historic periods. By clicking on a sub-event, additional information appears in the interface (i.e., all tweets, pictures and videos related to this sub-event). A filtering mechanism allows highlighting of sub-events containing specific keywords.

5. Online sub-event detection

For our detection process we identified two (online) processing steps: (i) real-time term addition and removal, (ii) online clustering to identify sub-events from streaming data. As a result of the dynamic evolution of an emergency situation, sub-events may evolve and vanish, and new sub-events may emerge over time. This evolution is dictated by the incoming data items collected from the social media platforms. That is, the vocabulary (i.e., used terms) for describing the incidents changes over time. Therefore, a dynamic indexing approach is needed to continuously track the evolution of terms.

We implemented three indexing approaches, including our own “Learn & Forget Model” indexing approach. For online processing, the following general steps performed after each batch of items can be identified (extended from [5]):

- Geo-tagging (see Section 6): Items from Flickr and YouTube are automatically tagged in order to identify the geo-coordinates (tweets are already tagged).
- Online indexing (for details see Section 7):
  - Standard incremental term frequency-inverse document frequency (tf-idf) as used in Brants et al. [46] and described in Section 7.1.
  - Skewness for detecting “bursty” keywords which are then used as terms/features for clustering. Skewness is one of the criteria used by Lee et al. [48] for bursty keywords detection (details in Section 7.2).
  - Our “Learn & Forget Model” ([5], Section 7.3). The terms own weights. These weights are used to identify important and non-important terms.
- Index augmentation:
  - By using the different selection mechanisms, incoming data items (i.e., retained terms from the previous step) are adopted into vector representations.
  - Geo-data are added to the term vector representations.
- Online clustering (see Section 8):
  - Existing cluster representations (from the previous step) are adapted based on the new term set (i.e., outdated ones are removed and new important terms are added).
  - The new data items of the batch are clustered.
- Visualization (see Fig. 6, Section 9):
  - Sub-events/clusters are labeled with the most important terms and visualized in a map of the web-based user-interface.
  - Browsing functionalities are offered to explore the results and filter additional information.
The next sections describe the geo-tagging and the indexing methods currently implemented in our framework. Additionally, the online clustering algorithm and its adaptation to the changing terms is described as well.

6. Geo-tagging

To identify the location of picture and video items, we use geo-tagging to tag new items by means of named entity recognition [56]. The location entities come with the textual metadata (i.e., title, description, and tags) of Flickr and YouTube. The first two locations (denoted as \( L_1 \) and \( L_2 \), e.g., Manhattan, NY) of an item are used to assign coordinates. If no location information is found in the textual metadata, the user location – if available – is used. If no location is found at all, the item is discarded from further processing.

We search first for the location in the text instead of taking the user location into account, because people can write about things independent of their home location, e.g., focusing on their current location.

Coordinates (related to the identified locations \( L_1 \) and \( L_2 \)) are searched by accessing the geographical database Geo-Names.org\(^1\). The provided GeoNames WebService offers the possibility to perform either a search within the database or through Wikipedia to extract/obtain coordinates.

To support geo-tagging, we also include a context \( \kappa \) into the search string (e.g., USA) to limit the problem of identifying cities with the same name in other countries outside the given context. In general, the context (e.g., the affected country or city) is known in an emergency situation. This could be changed on the fly if the context changes, i.e., the crisis shifts to another country or area.

The geo-tagging tries to determine the coordinates by performing the following processing sequence: (1) combine all identified locations plus the context to search for coordinates (i.e., \( \langle L_1, L_2, \kappa \rangle \)), (2) search for the first location string combined with the context \( \langle L_1, \kappa \rangle \), (3) search for the second location string combined with the context \( \langle L_2, \kappa \rangle \).

If it is not possible to identify the coordinates with the given option, in each step, an additional Wikipedia search with this option is performed. The coordinates can be extracted from the related Wikipedia entry. The corresponding item is annotated with the identified coordinates. If no coordinates are determined, the item is discarded from further processing.

7. Online indexing

There are already some approaches that take the evolution of terms over time into account. We do not distinguish between different data channels (e.g., between Twitter and Flickr) in calculating the indexing. The reason for that is that we want to identify sub-events based on the current incoming items, independently of any synchronization between those channels. We implemented the following indexing methods.

7.1. Incremental tf-idf

In general, one data item is represented as a vector of indexing terms reflecting on the importance of each term for that specific item. The traditional term frequency-inverse document frequency (tf-idf) [58] is used as follows:

\[
\text{idf}_t = \log \frac{N}{df_t}, \quad \text{tf}_t \cdot \text{idf}_t = \text{tf}_{t,d} \cdot \text{idf}_t,
\]

where \( \text{tf}_{t,d} \) shows the term frequency \( \text{tf}_{t,d} \) of term \( t \) in the document \( d \) times the inverse document frequency \( \text{idf}_t \) of all documents in the corpus containing \( t \). The \( \text{idf}_t \) of term \( t \) is calculated based on \( N \) representing all known items and \( df_t \) that represents the frequency of term \( t \) in the \( N \) documents.

In an incremental model the document frequencies are adapted or calculated based on new incoming terms. This could be done in two ways [46]. First, a training set is used to identify initial tf-idf values, which change as new documents arrive. Second, it is possible to compute tf-idf values from scratch using incoming documents. In our case, the creation of initial tf-idf values is often not feasible due to the unique characteristics (e.g., different impacts of a crisis due to the nature of the crisis, the affected location, and infrastructure) of an emergency situation and the missing training data. Hence, we implemented the second option where frequencies are calculated from scratch. This means \( N \) represents all items seen so far from the system and it continuously updates the \( df_t \) when new items arrive (i.e., incremental tf-idf) as suggested by Brants et al. [46].

Additionally, we remove terms having a very low document frequency (i.e., \( df_t < \theta_d \)) as they are known for not being very informative [46]. From the remaining terms, those with the highest tf-idf values are selected. All items from the current batch (period) are represented with the terms extracted from this step.

7.2. Skewness

“Bursty words” may help reveal indexing terms from incoming documents to identify sub-events. Lee et al. [48] describe a combined measure, including skewness, to identify such frequent terms in a period of one day. We do not implement all facets of the combined measure, e.g., as it also punishes periodic terms which, in our case, are not seen as irrelevant (e.g., reoccuring situations can have repeating terms, like fire and damage) and therefore should not be punished.

Skewness is a statistical measure which we use in this work to find suitable indexing terms for a period of time; it is calculated as follows [48]:

\[
\text{skewness}_{x,p} = \frac{E((x_{t,p} - \mu(x_{t,p}))^3)}{\sigma(x_{t,p})^3}
\]

The skewness of term \( t \) is calculated based on a vector \( x_{t,p} \) that contains all term frequencies of term \( t \) in all known documents \( \langle d_1, d_2, d_3, \ldots, d_k \rangle \) in the current period \( p \) (i.e., \( x_{t,p} = < \text{tf}_{1,p,d_1}, \text{tf}_{1,p,d_2}, \ldots, \text{tf}_{1,p,d_k} > \)). \( \mu \) and \( \sigma \) describe the mean and the standard deviation of \( x_{t,p} \). The more skewed the distribution function, the higher the values of \( \text{skewness}_{x,p} \) [48].

Here, we also remove terms that have a low document frequency (i.e., \( df_t < \theta_d \)). Afterwards, we select terms with the highest skewness for clustering. This gives us the most important indexing terms for the current period. For clustering, we transform the identified indexing term set to the vector space model representation. Hence, we calculate the tf-idf values for each period considering the selected terms.

7.3. Learn & Forget (L&F) term selection

This method implements a weighted version of the tf-idf. It uses a weighting mechanism for smooth removal of outdated terms and the inclusion of new ones. The tf-idf is calculated for a batch of documents (i.e., calculating \( N \) and \( df_t \)) based on a period \( p \) (given by the user). The period depends on the nature of the crisis

\[^{2}\text{Brants et al. [46] suggest a threshold of } \theta_d = 2.\]

Please cite this article as: D. Pohl, et al., Online indexing and clustering of social media data for emergency management, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.01.084
(e.g., slow or fast moving) and on the characteristics of the data stream (e.g., number of incoming items per time unit).

The additional weighting mechanism for calculating the relevance of each term is based on the incoming documents containing that specific term. We apply a first-order discrete time low pass filter [59, Eq. 8.62] in order to smooth incoming signals. The weights for known terms are refreshed based on Eq. (3a) and (3b), after each sampling interval $k_t$ at time $k$. The sampling interval $k_t$ has to be defined as a fraction of $p$, e.g., $k_t = 5$ min and $p = 30$ min.

$$\gamma_{t} = (1-\gamma) \cdot u_{t,k} + \gamma \cdot \gamma_{t,k-1}, \quad u_{t,k} > \gamma_{t,k-1} \quad \text{(3a)}$$
$$\gamma_{t} = (1-\delta) \cdot u_{t,k} + \delta \cdot \gamma_{t,k-1} \quad \text{otherwise} \quad \text{(3b)}$$

$\gamma_{t,k}$ indicates the weight of term $t$ at time $k$. $\gamma_{t,k-1}$ denotes the weight of the term $t$ from the previous sampling interval at time $k-1$. $u_{t,k}$ is the number of incoming documents containing the term $t$ at sampling time $k$. The first line (Eq. (3a)) of the equation serves to point out the novelty brought by the new items (the smaller $\gamma$, the faster the learning of incoming information). The second line (Eq. (3b)) is used to define the speed of forgetting the indexing terms (the higher $\delta$, the slower the forgetting).

$\gamma$ and $\delta$ are empirical values defined by the emergency manager based on his/her experiences (e.g., fast or slow evolving emergency). We suggest a ratio $\gamma < \delta$ which indicates that a high number of incoming items with term $t$ are learned faster than this term is being forgotten. Fig. 2 shows for different $\gamma$-$\delta$ settings the behavior of the weighting $g_{t,k}$ for incoming tweets $u_{t,k}$ at each timestamp $t$.

The resulting weights $g_{t,k}$ denote the importance of a term over time and act as memory to remember terms. The importance (Eq. (4)) is given by the ratio between the current valid weight and the maximum weight of the term $t$ reached during the selection phase.

$$\text{importance}_{t,k} = \frac{g_{t,k}}{g_{\max}} \quad \text{(4)}$$

Terms with the highest value based on Eq. (4) are identified as important and used for clustering in period $p$. Terms that have weights below a certain importance factor are removed from the possible term set. We empirically set this threshold to $\beta = 0.2$ meaning that 80% of the importance have been lost.

The weight is then included in the tf-idf formula (see Eq. (5)) to ensure the smooth removal of terms.

$$\text{scaled tf-idf}_{t,d} = \text{importance}_{t,k} \cdot \text{tf}_{t,d} \cdot \text{idf}_{t} \quad \text{(5)}$$

The $\text{scaled tf-idf}_{t,d}$ for item $d$ and term $t$ considers the importance of term $t$ (see Eq. (4)) at the end of each period $p$. The calculation of the weights is not done after each new item; instead, the incoming items are accumulated considering the sampling time $k$ and clustered after the period $p$.

The changed term set gained from all three indexing methods is also reflected in the clustering. This is done by deleting outdated terms from the clustering prototypes and by adding new ones that are relevant to the new batch.

8. Online clustering

Terms identified via the online indexing step are included in the online clustering algorithm after each period. We adapt the Growing Gaussian Mixture Models (2G2M) algorithm [6] to handle complete unlabeled data (see Algorithm 1). Clusters are described as multivariate Gaussians, where the number of Gaussians changes dynamically taking into account the number of terms in the incoming data. Because sub-event detection is done in an unsupervised way, meaning that social media items are not labeled, the detection algorithm should rely on clustering that does not require any pre-labeled data. Therefore, we adapted the online algorithm described in [6] to our case by removing the steps dealing with the case of labeled data. The following symbols are used in Algorithm 1 [6]:

**Algorithm 1.** Steps of 2G2M handling unlabeled data (adapted from [6])

1: Given a new input $x_i$, compute the probability of match of the input with each cluster: $\forall j = 1...K$

$$p_j = \begin{cases} \tau_j f_j(x_i; \mu_j, \Sigma_j) & \text{if } d_{ij}(x_i, \mu_j) < T_{\Sigma} \\ 0 & \text{otherwise} \end{cases} \quad \text{(6)}$$

$$d_{ij}(x_i, \mu_j) = \sqrt{\sum_{t=1}^{T} (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)} \quad \text{(7)}$$

2: Let $R = \{j \mid p_j > 0\}$ be the index set of Gaussians matching the input

3: createGaussian$\rightarrow (R = \emptyset)$ (i.e., no match found: $\emptyset, p_j > 0$)

4: if not createGaussian then

5: Compute the index of best matching Gaussian:

$$w = \arg \max_j p_j \quad \text{for } j = 1...K \quad \text{(8)}$$

6: Compute the expected posterior:

$$q_w = \sum_k p_{wk} \quad \text{(9)}$$

7: Update the parameters of the Gaussian:

$$c_w = c_w + q_w \quad \text{(10)}$$

$$\tau_w(t) = (1-\alpha)\tau_w(t-1) + \alpha q_w \quad \text{(11)}$$

$$\eta_w = \eta_w + \frac{1-\alpha}{c_w + \alpha} \quad \text{(12)}$$

$$\mu_w(t) = (1-\eta_w)\mu_w(t-1) + \eta_w x_i \quad \text{(13)}$$

$$\Sigma_w(t) = (1-\eta_w)\Sigma_w(t-1) + \eta_w (x_i - \mu_w(t-1))^2 \quad \text{(14)}$$

8: end if

9: if createGaussian then

10: Decay the weight of all Gaussians

$$\forall j = 1...K, \tau_j(t) = (1-\alpha)\tau_j(t-1) \quad \text{(15)}$$

11: Remove the least contributing Gaussian and create a new one initialized with the new input:

$$m = \arg \min_j (\tau_j) \quad \text{(16)}$$

$$c_m = c_m + q_m \quad \text{(17)}$$

$$\eta_m = \eta_m + \frac{1-\alpha}{c_m + \alpha} \quad \text{(18)}$$

$$\mu_m(t) = (1-\eta_m)\mu_m(t-1) + \eta_m x_i \quad \text{(19)}$$

$$\Sigma_m(t) = (1-\eta_m)\Sigma_m(t-1) + \eta_m (x_i - \mu_m(t-1))^2 \quad \text{(20)}$$
Algorithm 1 starts by comparing the input $x$ with all currently known clusters in the system. A similarity test between the input and the corresponding Gaussians of a cluster is performed using the Mahalanobis distance (Algorithm 1, Step 1: $d(x)$ in Eq. (7)). If the distance is below the predefined threshold $T_{\Sigma}$, the corresponding cluster is considered as similar to the current input (Algorithm 1, Step 1: $p_i$).

If many clusters are identified, the index of the most similar Gaussian (i.e., so-called “winner takes all” approach) is retained (Algorithm 1, Step 5). Afterwards, the posterior probability of the winning cluster is calculated and the parameters of this cluster (e.g., weight of the cluster $\phi_j$, covariance matrix $\Sigma_j$, mean value $\mu_j$) are adapted based on the input values (Algorithm 1, Steps 6 and 7). If no matching cluster could be identified, a new Gaussian/cluster is created using the current input (Algorithm 1, Step 11). Before the new Gaussian is created, the weights of the remaining Gaussians are decayed and the least contributing Gaussian is removed.

The algorithm considers also the merge and split of existing Gaussians to refine the created model (Algorithm 1, Steps 13–15). During clustering, very close clusters are merged. The decision is based on the Kullback–Leibler divergence (see Eq. (19)) [6,60]. If the divergence is below a predefined threshold $T_{\text{merge}}$, they are merged. On the other hand, large clusters are split. A split is performed when the volume in Eq. (18) exceeds a certain threshold $T_{\text{split}}$. Clusters are split along their dominant principal components [6].

As the indexing terms can change after each batch (period), the clusters have to be adapted. The adaptation to the new term set is performed by updating the Gaussian parameters $\mu_j$ and $\Sigma_j$ of cluster $j$. When removing terms, the corresponding values (rows/columns) of these old terms are removed. If there are new terms to consider, zero values for $\mu_j$ and $\Sigma_j$ are included for each cluster $j$.

### 9. Experiments

For the sake of evaluation, we use real-world data (i.e., Tweets, Flickr and YouTube items) from the Hurricane Sandy 2012. In

![Fig. 3. Distribution of used media items (visualized via OpenStreetMap).](image-url)

<table>
<thead>
<tr>
<th>Method</th>
<th>Setting</th>
<th>DB</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(\gamma, \delta)$</td>
<td>60</td>
<td>110</td>
</tr>
<tr>
<td>L&amp;F</td>
<td>(0.2, 0.6)</td>
<td>6.532</td>
<td>9.087</td>
</tr>
<tr>
<td></td>
<td>(0.5, 0.6)</td>
<td>6.524</td>
<td>8.033</td>
</tr>
<tr>
<td></td>
<td>(0.2, 0.4)</td>
<td>5.876</td>
<td>7.771</td>
</tr>
<tr>
<td>Skewness</td>
<td>inc. tf-idf</td>
<td>5.805</td>
<td>8.067</td>
</tr>
<tr>
<td></td>
<td>inc. tf-idf</td>
<td>5.805</td>
<td>8.327</td>
</tr>
</tbody>
</table>

Table 1

Topics of Hurricane Sandy 2012 given in Wikipedia [61].

<table>
<thead>
<tr>
<th>Topics</th>
<th>Description (period: 29th October, local time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airports/Flights</td>
<td>Closed airports and canceled flights, i.e., JFK, Newark (8 PM)</td>
</tr>
<tr>
<td>Evacuation</td>
<td>Several hospitals and FDNY Emergency Medical Services were evacuated</td>
</tr>
<tr>
<td>Flooding</td>
<td>Different districts, in addition also tunnels (7 PM) and sub-ways</td>
</tr>
<tr>
<td>Power/electricity</td>
<td>Power outages in several districts, e.g., Manhattan, Queens, Staten Island, Brooklyn, Bronx, etc.</td>
</tr>
<tr>
<td>Fire</td>
<td>Several fires due to fallen trees or blown-up transformers, e.g., Breezy Point, Queens (approx. 11 PM)</td>
</tr>
<tr>
<td>Wind</td>
<td>Fallen trees and broken branches, damage due to the heavy wind, crane collapsed, etc.</td>
</tr>
</tbody>
</table>

Please cite this article as: D. Pohl, et al., Online indexing and clustering of social media data for emergency management, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.01.084
In particular, we collected items during the main impact of the hurricane from October 29, 2012, till November 1, 2012.

Considering the enormous number of tweets, approximately 3.7 million tweets, we conducted our experiments based on a smaller snapshot to evaluate the approach for its effectiveness. Therefore, we considered only geo-tagged tweets related to the locations around Manhattan, Brooklyn, and New Jersey. This results in 1003 tweets for the period from October 29, 02:00 pm to October 30, 02:30 am. Beside the tweets, we also used 286 pictures from Flickr and 167 videos from YouTube. In total, we have a data set consisting of 1456 social media items distributed on the three locations as shown in Fig. 3.

In Table 1, the major topics related to the Hurricane Sandy 2012 are extracted from the report given in Wikipedia [61]. Topics are therefore readily available and are considered to be the reference topics/sub-events. They will be compared against those detected by our sub-event detection approach. The major incidents during the hurricane are related to damages, flooding, and power outages. They can also be found in the results obtained via the tested algorithms (see Section 9.2, Tables 3–5).

9.1. Description

We evaluated different online/indexing methods as described in Section 7. In particular, we compare our approach, the Learn & Forget method (Section 7.3, Pohl et al. [5]), against the incremental tf-idf (Section 7.1, Brants et al. [46]) and the skewness-based method (Section 7.2, Lee et al. [48]). Moreover, different settings (time and parameter) of the Learn & Forget method are tested.

We consider only nouns (e.g., flood and damage) as possible item candidates. The extracted nouns are stemmed using the common Porter Stemmer [62]. Similar nouns are grouped together and treated as one (e.g., US and USA, flood and flooding). This is performed by making use of the relations between concepts managed in WordNet [63]. The resulting terms are then weighted.
Table 3
Resulting topics based on the cluster IDs; includes the stem forms of words and markers for important words describing sub-events.

<table>
<thead>
<tr>
<th>ID</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L&amp;F ({\gamma = 0.5, \delta = 0.6}; 60 \text{ terms})</td>
</tr>
<tr>
<td>2</td>
<td>new park apocalyps con sandi hurricane</td>
</tr>
<tr>
<td>3</td>
<td>new ny sandi frankenstorm hurricane york power</td>
</tr>
<tr>
<td>4</td>
<td>ny mondai new frankenstorm sandi con hurricane</td>
</tr>
<tr>
<td>5</td>
<td>hurricane apocalyps new frankenstorm octob sandi citi</td>
</tr>
<tr>
<td>6</td>
<td>hurricane new citi sandi ny other nyc</td>
</tr>
<tr>
<td>7</td>
<td>york nyc park build manhattan sandi fire</td>
</tr>
<tr>
<td>8</td>
<td>new nyc apocalyps explos sandi power station</td>
</tr>
<tr>
<td>9</td>
<td>hurricane other power explos ed part station</td>
</tr>
<tr>
<td>10</td>
<td>amb center evacu am ambul chri edt</td>
</tr>
<tr>
<td>11</td>
<td>wind octob new sandi hurricane york other</td>
</tr>
<tr>
<td>12</td>
<td>sandi citi apocalyps power york con hurricane</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Brants ({60 \text{ terms})</td>
</tr>
<tr>
<td>1</td>
<td>hurricane sandi dai tunnel power fall manhattan</td>
</tr>
<tr>
<td>2</td>
<td>apocalyps frankenstorm sandi other hurricane</td>
</tr>
<tr>
<td>3</td>
<td>newyork nyc park ny sandi storm hurricane</td>
</tr>
<tr>
<td>4</td>
<td>wind crane us storm time build central</td>
</tr>
<tr>
<td>5</td>
<td>street hurricane dai east bridge video power</td>
</tr>
<tr>
<td>6</td>
<td>mondai citi sandi hurricane york new power</td>
</tr>
<tr>
<td>7</td>
<td>nyc citi park mondai evacu new manhattan</td>
</tr>
<tr>
<td>8</td>
<td>sandi hurricane citi york new home jersie</td>
</tr>
<tr>
<td>9</td>
<td>nyc east hurricane</td>
</tr>
<tr>
<td>10</td>
<td>hudson water flood river east coast hit</td>
</tr>
<tr>
<td>11</td>
<td>nyc wind video sandi manhattan us hurricane</td>
</tr>
<tr>
<td>12</td>
<td>rain hurricane sandi ny new us york</td>
</tr>
<tr>
<td>13</td>
<td>other apocalyps frankenstorm ny sandi york hurricane</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lee ({60 \text{ terms})</td>
</tr>
<tr>
<td>1</td>
<td>hurricane sandi dai tunnel power fall manhattan</td>
</tr>
<tr>
<td>2</td>
<td>newyork nyc park ny sandi storm hurricane</td>
</tr>
<tr>
<td>3</td>
<td>crane new us storm park time central</td>
</tr>
<tr>
<td>4</td>
<td>nyc sandi hurricane wai york new street</td>
</tr>
<tr>
<td>5</td>
<td>street hurricane dai east manhattan power wai</td>
</tr>
<tr>
<td>6</td>
<td>citi hurricane</td>
</tr>
<tr>
<td>7</td>
<td>nyc oct citi park mondai wai ey</td>
</tr>
<tr>
<td>8</td>
<td>oct mondai sandi hurricane york new power</td>
</tr>
<tr>
<td>9</td>
<td>other apocalyps frankenstorm ny sandi york new</td>
</tr>
<tr>
<td>10</td>
<td>hurricane dai peopi sandi ny new york</td>
</tr>
<tr>
<td>11</td>
<td>apocalyps frankenstorm sandi other hurricane</td>
</tr>
<tr>
<td>12</td>
<td>nyc east hurricane</td>
</tr>
<tr>
<td>13</td>
<td>nyc citi jersie flood east coast manhattan</td>
</tr>
<tr>
<td>14</td>
<td>frankenstorm apocalyps other sandi ny new york</td>
</tr>
<tr>
<td>15</td>
<td>nyc wind video sandi manhattan us hurricane</td>
</tr>
</tbody>
</table>

We evaluate the clustering using the Silhouette (S) [64] and the Davies–Bouldin (DB) [65] metrics. The S metric identifies how close related items in a specific cluster are. High values of the Silhouette metric indicate good and well-separated clustering.

The Silhouette is expressed as follows for an item \(m\) and a cluster \(i\) [64]:

\[
\begin{align*}
D_m &= \min_{j \neq i} d_{mj}, \\
S_m &= \frac{b_m - a_m}{\max(a_m, b_m)}
\end{align*}
\]

\(d_m\) represents the average dissimilarity of the item \(m\) to all other items in the same cluster \(i\). \(d_{mj}\) describes the average dissimilarity of all items from the other clusters \(j\) to \(m\). \(b_m\) represents the smallest dissimilarity of \(m\) to \(d_{mj}\). The S value of a cluster is the average of \(S_m\) from each item \(m\) in the cluster.

The DB index is expressed as follows [65]:

\[
DB = \frac{1}{n} \sum_{i=1}^{n} R_i
\]

where \(R_i = \max_{j=1,...,n, j \neq i} \frac{S_i + S_j}{d_{ij}}\) \(i = 1, ..., n\)

The DB index describes the similarity between clusters based on the dispersion \(S_i\) of a cluster \(i\) and by considering \(d\) as dissimilarity measure between two clusters [65]. Small values of DB indicate a good clustering emphasizing that clusters are dissimilar to each other.

The parameters for the clustering algorithm (shown in Algorithm 1) are empirically evaluated and set for all three indexing methods to \(\alpha = 0.01, K = 1000, T_e = 3, T_{merge} = 8, T_{split} = 20\).

9.2. Comparison of the results

In all experiments, the number of terms retained is the same for all methods. Two experiments (60 terms, 110 terms) have been performed. For the Learn & Forget (L&F) method, several settings have been tested. Table 4 shows the average DB and S values over all periods \((p = 30 \text{ min})\).

In addition, Figs. 4 and 5 outline the behavior of the different methods when data comes over time for different settings. The

Table 4
Resulting topics based on the cluster IDs; includes the stem forms of words and markers for important words describing sub-events.

<table>
<thead>
<tr>
<th>ID</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L&amp;F ({\gamma = 0.2, \delta = 0.4}; 60 \text{ terms})</td>
</tr>
<tr>
<td>1</td>
<td>wind york park build part sandi fire</td>
</tr>
<tr>
<td>2</td>
<td>hurricane apocalyps new frankenstorm sandi york manhattan</td>
</tr>
<tr>
<td>3</td>
<td>new nyc part explos sandi manhattan station</td>
</tr>
<tr>
<td>4</td>
<td>new hurricane sandi nyc park citi</td>
</tr>
<tr>
<td>5</td>
<td>hurricane manhattan power explos ed station evacu</td>
</tr>
<tr>
<td>6</td>
<td>octob con new sandi hurricane york power</td>
</tr>
<tr>
<td>7</td>
<td>mondai abc center ambul chri edt</td>
</tr>
<tr>
<td>8</td>
<td>sandi nyc citi manhattan york hurricane</td>
</tr>
</tbody>
</table>

Table 5
Resulting topics based on the cluster IDs; includes the stem forms of words and markers for important words describing sub-events.

<table>
<thead>
<tr>
<th>ID</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L&amp;F ({\gamma = 0.2, \delta = 0.4}; 110 \text{ terms})</td>
</tr>
<tr>
<td>1</td>
<td>park ny other brooklyn coast sandi evacu</td>
</tr>
<tr>
<td>2</td>
<td>citi explos other power ny sandi york</td>
</tr>
<tr>
<td>3</td>
<td>york park build video con part evacu</td>
</tr>
<tr>
<td>4</td>
<td>other coast video frankenstorm part sandi storm</td>
</tr>
<tr>
<td>5</td>
<td>hurricane apocalyps other sandi east york manhattan</td>
</tr>
<tr>
<td>6</td>
<td>new storm hurricane frankenstorm apocalyps ed station</td>
</tr>
<tr>
<td>7</td>
<td>nyc other bithe alaneelieeep boss apocalyps</td>
</tr>
<tr>
<td>8</td>
<td>manhattan abc center ambul chri edt</td>
</tr>
<tr>
<td>9</td>
<td>citi other mondai explos power sandi apocalyps</td>
</tr>
<tr>
<td>10</td>
<td>manhattan abc center ambul chri edt</td>
</tr>
<tr>
<td>11</td>
<td>other citi apocalyps power sandi video coast</td>
</tr>
<tr>
<td>12</td>
<td>hurricane apocalyps other sandi video mondai power</td>
</tr>
<tr>
<td>13</td>
<td>street octob bit asi condado other sandi</td>
</tr>
<tr>
<td>14</td>
<td>other coast 72nd halloween st frankenstorm instacolfag</td>
</tr>
<tr>
<td>15</td>
<td>wind nyc storm water flood lower attack</td>
</tr>
<tr>
<td>16</td>
<td>manhattan abc center ambul chri</td>
</tr>
<tr>
<td>17</td>
<td>high other lic dry gotta laugh apocalyps</td>
</tr>
</tbody>
</table>

Please cite this article as: D. Pohl, et al., Online indexing and clustering of social media data for emergency management, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.01.084
methods exhibit little variation when using different clustering evaluation measures, in the case of DB and S. Therefore, changing the measure does not imply any clear response trend of the algorithms.

Table 2 summarizes the metrics for different term settings (60 and 110) indicating the best results for each term setting in bold. By increasing the number of terms from 60 to 110, the results for the L&F improve (see Table 2, column 110 terms). In general, all three methods show very similar outcomes (see also the DB and S values in Figs. 4 and 5), but differ in the way features are selected (see Table 3 [60 terms]).

Tables 3–5 show the results of the different approaches and settings. Terms are given in their stem form that the natural language processor (NLP) outputs. To support their interpretation, the stems can be mapped onto the topics' description in Table 1. Bold terms indicate important terms identified by one approach only. Italic terms indicate terms identified by all approaches. In the following sections, the most important settings are compared with each other and the differences are highlighted.

Given Table 3, the L&F method identifies additional terms and hence uncovers specialized sub-events (topics) compared to the other approaches. For example, for the period 00:30 local time (05:30 UTC) on October 30 (this period shows also small DB values in Fig. 4(a)), information regarding fire, evacuation and explosion was extracted (see Table 3). The “explosion” was only labeled by the L&F method, also information regarding fire services was only extracted by our approach. The incremental tf-idf approach identified a topic concerning the evacuation. Skewness identifies none of these concepts, but together with the inc. tf-idf an additional item documenting the flooding in Lower Manhattan is detected. Information regarding power outages or comments on power supply are covered by all approaches (“power” in Table 3). Additionally, the L&F method achieves a smaller number of sub-events. A visualization of the sub-events identified by L&F can be found in Fig. 6. It also identifies topics regarding canceled flights in another period at 02:00 UTC (not shown in this table).

The L&F for the \((\gamma = 0.2, \delta = 0.4)\) setting shows that the terms are learned and forgotten very fast (see Table 4). This results in a peak in the DB diagram, as the terms in the term set are changing fast too (especially for specific periods). The L&F with \(\gamma = 0.2\) and \(\delta = 0.4\) also identifies important terms, but aggregates items too strong/tight.

When increasing the number of terms as the Information Retrieval (IR) literature suggests, we noticed that the L&F method increases its performance. Results for the increased number of terms (110 terms) can also be found in Table 2. Details on the values of the metrics for each period can be found in Figs. 4 and 5 (on the right-hand side). It can be seen that for a high number of terms the L&F approach improves its performance. The L&F method \((\gamma = 0.2, \delta = 0.4)\) shows the best performance as more descriptive terms are included. More clusters are generated as additional information becomes available. For the period 05:30 a topic regarding the sub-event “flood” is identified (see Cluster 15 in Table 5). In summary, 17 sub-events are uncovered (as shown Table 5) compared to the 8 sub-events extracted with 60 terms in Table 4.

Table 2 shows that for an increased number of terms (from 60 to 110 terms) the L&F \((\gamma = 0.2, \delta = 0.4)\) approach improves performance compared to the other approaches. The DB value declines whereas the S value increases. Based on the DB and S results given in Tables 2 and 5 the setting \([\gamma = 0.2, \delta = 0.4]; 110\) terms provides the best results in this study.
9.3. Discussion and future work

The online indexing methods show similar behavior as can be seen in Figs. 4 and 5. They identify in general important sub-events related to flights, damage, flood, evacuation, power and different relief units. These major topics can also be found in Table 1. Increasing the number of terms shows also an increased performance for the L&F method (see Table 2). This method has advantages as it does not need to store all terms found in the data stream (see incremental tf-idf).

Irrelevant terms are removed from the index set. In addition, the L&F method allows us to control learning and forgetting of the terms by means of the parameters. This makes the method flexible for adapting to new situations (e.g., fast or slow changes) that can emerge during a crisis. Additionally, it allows us to memorize terms over periods. In contrast, the skewness method cannot identify easily terms over different periods.

In the future, we aim to devise an automatic method to adjust parameters for the clustering algorithm (i.e., recognizing conceptual drifts in the incoming information, the nature of the stream, etc.). It is also possible to incorporate visual features, especially from pictures, and extend the algorithm to handle those features appropriately.

10. Conclusion

This paper presents a framework for identifying sub-events based on crisis-related data. The identification uses a dynamic indexing and an online clustering algorithm. We investigated three online indexing methods (incremental tf-idf, skewness and Learn & Forget). The experiments show that for a higher number of indexing terms the Learn & Forget method performs better than the incremental tf-idf and skewness methods. Sub-events related to important incidents can be identified, like power outage, flood, evacuation, etc. A demonstration performed in September 2013 also showed the usefulness of the suggested framework during a real-time emergency response exercise. In the future, we will focus on an automatic method to identify and handle concept drifts in the data, e.g., by re-initializing the algorithm or by adjusting the Learn & Forget parameters.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007–2013) under Grant agreement no. 261817 and was partly performed in the Lakeside Labs research cluster at Alpen-Adria-Universität Klagenfurt.

References


Daniela Pohl received her Dipl.-Ing. (Master’s degree) in Computer Science in 2008 at the Alpen-Adria-Universität Klagenfurt, Austria. She is currently a research assistant and Ph.D. candidate at the Institute of Information Technology, Alpen-Adria-Universität Klagenfurt. She works in the scope of the EU-funded FP7 project BRIDGE (www.bridgeproject.eu) to develop technical solution to improve crisis and emergency management. Her research interests include social media analysis, information retrieval, data mining, and machine learning.

Abdelhamid Bouchachia is currently an Associate Professor at Bournemouth University, Department of Computing, Smart Technology Research Centre, UK. His major research interests include Machine Learning and Soft Computing with a particular focus on online/incremental learning, semi-supervised learning, predictive systems, and uncertainty modeling. He is the general chair of the International Conference on Adaptive and Intelligent Systems (ICAIS). He serves as a program committee member for many conferences. He also serves as an Associate Editor of Evolving Systems and acts as a member of Evolving Intelligent Systems (EIS) Technical Committee (TC) of the IEEE Systems, Man and Cybernetics Society, the IEEE Task-Force for Adaptive and Evolving Fuzzy Systems and the IEEE Computational Intelligence Society.

Hermann Hellwagner is a full professor of Informatics in the Institute of Information Technology (ITEC), Klagenfurt University, Austria, leading the Multimedia Communications group. His current research areas are distributed multimedia systems, multimedia communications, and quality of service. He has received many research grants from national (Austria, Germany) and European funding agencies as well as from industry, is the editor of several books, and has published more than 200 scientific papers on parallel computer architecture, parallel programming, and multimedia communications and adaptation. He is a senior member of the IEEE, member of the ACM, GI (German Informatics Society) and OCG (Austrian Computer Society), and Vice President of the Austrian Science Fund (FWF).