Online Processing of Social Media Data for Emergency Management

Daniela Pohl
Institute of Information Technology
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
daniela@itec.uni-klu.ac.at

Abdelhamid Bouchachia
Smart Technology Research Center
Bournemouth University
Bournemouth, UK
abouchachia@bournemouth.ac.uk

Hermann Hellwagner
Institute of Information Technology
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
hellwagn@itec.uni-klu.ac.at

***Abstract*—Social media offers an opportunity for emergency management to identify issues that need immediate reaction. To support the effective use of social media, an analysis approach is needed to identify crisis-related hotspots. We consider in this investigation the analysis of social media (i.e., Twitter, Flickr and YouTube) to support emergency management by identifying sub-events. Sub-events are significant hotspots that are of importance for emergency management tasks. Aiming at sub-event detection, recognition and tracking, the data is processed online in real-time. We introduce an incremental feature selection mechanism to identify meaningful terms and use an online clustering algorithm to uncover sub-events on-the-fly. Initial experiments are based on tweets enriched with Flickr and YouTube data collected during Hurricane Sandy. They show the potential of the proposed approach to monitor sub-events for real-world emergency situations.**

***Keywords*-Online Clustering, Sub-Event Detection, Crisis Management**

I. Introduction

Sharing of information and knowledge is a key point during emergency response. However, often it is not possible for relief units to gather this information. Due to the popularity of the mobile Internet and the broad acceptance of social media (e.g., Twitter) as a communication channel, social media offers the opportunity to collect useful information. Also, different studies (e.g., [1]) show the significance of social media in emergencies.

An intelligent approach is needed to support professional first responders in managing and analyzing this data. For this purpose, we focus on the detection of sub-events. Sub-events are related to topics in "topic detection and tracking" (TDT) where real-world events have to be identified/followed [2]. Identified sub-events describe situations where emergency management is demanded.

Our previous work tested clustering algorithms for their suitability in detecting sub-events retrospectively [3]. For the identification of sub-events, offline clustering algorithms are investigated. These algorithms cannot be directly deployed in real-time during an emergency. Thus, we investigate online sub-event detection. Specifically, we rely on online feature selection and real-time clustering. We consider Twitter, Flickr, and YouTube data. We use terms extracted from the given messages (text fields) as features for the clustering. We follow a batch-based approach to process the new incoming messages. The batches are temporal windows with a parameterized length (duration). Each batch is processed as a collection of documents by extracting and filtering terms.

The structure of the paper is as follows. Section 2 discusses related work. Section 3 addresses the terminology regarding topic detection and tracking. Section 4 outlines the online/incremental feature selection and clustering algorithm. Section 5 shows the details of the online feature selection based on a learning and forgetting model. Section 6 describes the online clustering algorithm. In Section 7, the experimental setting and the results are presented. Section 8 concludes the paper.

II. Related Work

The present work relates to "topic detection and tracking" where the goal is to identify/track topics based on incoming data. In general, there is a focus on Twitter for social media analysis. Most of the approaches operate with auxiliary material or in a static manner. Osborne et al. [4] describe online story detection based on Twitter which uses Wikipedia to verify the identified stories. Chakrabarti and Punera [5] describe an approach for summarizing already detected events. Twitcident [6], is based on predefined keywords or manually inserted rules. CrisisTracker by Rogstadius et al. [7] (based on [8]) represents a crowdsourcing tool to support volunteers in processing incoming messages. Shen et al. [9] show event-detection based on general important concepts (e.g., names of persons and organizations).

Natural language processing of tweets produces a huge number of indexing terms/features. To deal efficiently with them, feature selection methods are used. Within clustering, the selection is based on the structure of the data, e.g., document frequency, as there are no class labels. Examples for feature selection mechanisms in the clustering context can be found in [10], [11] and [12].

With a weighting mechanism in the feature selection the changes in streaming data during emergency management can be modeled. Most of these weighting mechanisms can be found in the classification area. Bouchachia and Mittermeir [13] describe the weighting of features for fuzzy classification. This approach is similar to our idea, but we

use a different weighting function to select features on the fly. Also [14] and [15] describe feature selection in the classification context. However, the exact classes (i.e., labels) are usually not known in advance during an emergency.

We focus on clustering methods, which do not need labeled data or additional effort for preparing the analysis, e.g., initial/training data. Additionally, we consider a weighting mechanism for a smooth feature selection and reduction. Our goal is to identify important sub-events by grouping incoming data according to these sub-events.

III. Events and Sub-Events

Emergency-related sub-events comprise different incidents of a crisis, e.g., flooding, damages or power outage in different districts. An event (i.e., the crisis) is described by its time and location, e.g., Hurricane Sandy in the USA in 2012. Sub-events are incidents originating in the context of the parent event, like flooding in Manhattan. Therefore, a sub-event is part of the parent event. A sub-event is represented or identified by information describing the specific incident, e.g., reports or postings of people.

This is similar to the definition used for "topic detection and tracking" (TDT) [2]. Specifically, a topic consists of different incidents (sub-topics) that are triggered by the topic. For consistency, we use a more specific emergency-based terminology, where we focus on events (topics) and sub-events (sub-topics). The main differences to TDT is the (fast) evolving situation of a disaster, the limited textual description that originates from social media compared to articles and the impossibility of labeling the data due to the unclear situation. We also have a more spatial-temporal focus on the data.

IV. Online Sub-Event Detection

To detect sub-events from streaming data we use two processing steps: (i) dynamic feature selection for adding and removing features on the fly and (ii) online clustering for discovering sub-events from streaming data.

Due to the high dynamics of a crisis, sub-events can evolve, vanish, or new ones emerge. Such evolution is reflected by the vocabulary used in messages, which changes over time. Dynamic feature selection aims at representing the messages' contents based on the most important terms for our features. Having the term vectors, online clustering is applied to uncover the sub-events. We have different generic steps for our batch mode processing:

- **Geo-tagging for Flickr and YouTube:** Due to the sparsity of geo-coordinates, we automatically tag the messages. Note that tweets are already tagged.
- **Online feature selection** ("learn & forget", see Section V): Terms get weights assigned and go through a selection process to discard non-important terms.
- **Message representation** (see Section V): By using an extended tf-idf formula, messages (actually the retained terms from the previous step) are reduced (compressed) into vectors.
- **Online clustering** (see Section VI):
  - Clusters are adapted based on the new feature set (outdated features are removed and new ones are added) related to the new batch of data.
  - Geo-data are added to the term vectors (see Section VI).
  - The new messages are clustered.
- **Visualization of the clustering results** (see Fig. 3, Section VII).

The weighting mechanism is used as memory to recall terms over time. The more often a term is used, the more important it is considered in the next clustering step and the longer it should be remembered. Each clustering step is incrementally performed on the new set of inputs (tweets, pictures, or videos). Further details are given in Sections V and VI.

V. Online Feature Selection: Learn & Forget

Messages are represented as vectors of weights that indicate how important a term in a message is. The common *term frequency-inverse document frequency* (tf-idf) [16] is used:

$$idf_t = log\frac{N}{df_t}, \qquad tf\_idf_{t,d} = tf_{t,d} \cdot idf_t \tag{1}$$

In Eq. 1 $tf\_idf_{t,d}$ shows the term frequency $tf_{t,d}$ of term t in the document d times the inverse document frequency idf_t of all documents in the corpus containing t. The idf_t of term t is calculated based on N representing all known messages and df_t describing all messages of N containing term t.

In an online setting, this approach cannot be directly adopted due to the ongoing arrival of new messages. For this purpose, an incremental tf-idf needs to be used. As we do not have training data for an initial model nor we want to privilege new arriving terms based on their small df_t [17], we aim at introducing another approach.

For a batch (period), we calculate the tf-idf (i.e., N and df_t) based on a period p (given for example by the user). The period is defined based on the characteristics of the crisis (slow or fast moving crisis).

For identifying the most important terms, we use a weighting mechanism to specify the relevance of the terms based on incoming documents (see Eq. 2). A first-order discrete time low pass filter inspired by the signal processing literature [18, Eq. 8.62] is used, which smoothes the incoming signals. The weights for known terms are refreshed based on Eq. 2 after a sampling interval k_s at time k. The sampling interval has to be a fraction of p, e.g., 5 minutes.

$$g_{t,k} = \begin{cases} (1-\gamma) \cdot u_{t,k} + \gamma \cdot g_{t,k-1} & u_{t,k} > g_{t,k-1} \\ (1-\delta) \cdot u_{t,k} + \delta \cdot g_{t,k-1} & \text{otherwise} \end{cases} \tag{2}$$

Here, $g_{t,k}$ denotes the weight of term t at time k. In the equation, $u_{t,k}$ refers to the incoming number of documents containing the term for the current sampling time k and $g_{t,k-1}$ describes the weight of the term t from the previous sampling at time $k-1$. The first formula describes how fast the term should be "learned" (the smaller γ, the faster the learning) and the second describes the behavior when "forgetting" the information (the higher δ, the slower the forgetting). Fig. 1 exemplifies the behavior for different settings, i.e., considering different "forgetting" factors.

The factors γ and δ are empirical values based on the experience of the emergency manager and the evolution (fast or slow) of the crisis. We suggest a ratio $\gamma < \delta$ which indicates that a high number of incoming documents with term t is learned faster (weighted higher) compared to the forgetting of this term.

The weights give the possibility to calculate the importance of a term over different periods and act as memory (see Fig. 1). Term weights that are below a predefined importance factor ($\beta = 0.2$) are removed from the term set. The importance (see Eq. 3) is calculated as the ratio between the current weight of the term t and the maximum weight of this term reached during the application. This normalization ensures the comparability of the weights of different terms.

$$importance_{t,k} = g_{t,k} / g\_max_t \tag{3}$$

Based on the term weights, the most important terms are identified. This means, terms with the highest value based on Eq. 3 at the end of each period p are used for clustering. Hence, the term set changes due to the weighting mechanism and the drop-out of out-dated terms. The weighting is also included in our tf-idf calculation (see Eq. 4) to ensure the smooth removal of terms.

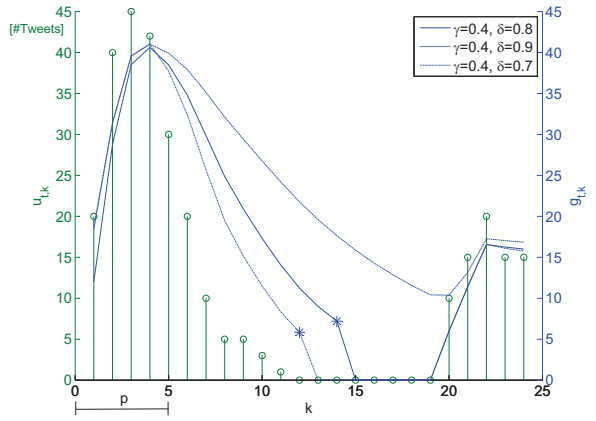


Figure 1. $g_{t,k}$ based on $u_{t,k}$ (e.g., $p = 5min, k_s = 1min$) with different learning and forgetting factors; removed features are marked with a $\star$

$$scaled\_tf\_idf_{t,d} = importance_{t,p} \cdot tf_{t,d} \cdot idf_t \tag{4}$$

The $scaled\_tf\_idf_{t,d}$ for message d and term t is calculated based on its importance (see Eq. 3) at the end of each period p. The calculation of the weights is not performed on each arriving message; rather, arriving messages are accumulated based on the sampling time k and clustered at time p.

The changing term set must be considered in the clustering too, by creating a mapping between the new feature set and the existing cluster centers/prototypes. This is performed by deleting features from the prototypes, which do not heavily occur in the batch, and adding new features, which appear often in the new batch. The new added features to the cluster prototypes are set to zero.

VI. Online Clustering

The identified features are fed into the online algorithm after each period. Therefore, we adopted the *Growing Gaussian Mixture Models* (2G2M) algorithm (for details see [19]). Each cluster is described via a multivariate Gaussian, where the dimensions are given by the number of current features available to the algorithm. We adapted the algorithm in a way that it handles complete unlabeled data.

Calculating the probability of a match between the input and the corresponding Gaussians of a cluster is performed via the Mahalanobis distance [19]. If the value is beyond the predefined threshold τ_σ the input is seen as similar to the cluster.

Large clusters are split into two smaller clusters. The decision of a split is performed using the volume of the cluster. That is, if the volume of a cluster is above a threshold T_{split}, the cluster is split according to its dominant principal components [19].

Using the merge operation, two very similar/close clusters can be merged. The Kullback-Leibler divergence for multivariate Gaussians is used [19]. If the divergence for two clusters is below the threshold T_{merge}, they get merged.

As terms change over each period, the existing clusters have to be adapted to the new situation. Based on the feature selection described in Section V, the old terms are removed and new ones are added based on their importance. Terms lose their importance over time when the number of incoming documents with term t decreases. This decreases the importance/relevance of the term. The term is getting less and less important for the clustering result. If it is below a threshold, the term is removed from the term set. The inclusion of the weights results in a smooth drop-out of outdated terms, which creates room for new terms.

The adaptation of the clusters based on the evolution of the features can be performed through the Gaussian parameters. The old features (representing the clusters) are removed from the mean μ_j and covariance matrix Σ_j describing cluster j by removing the corresponding values and the rows/columns of the old feature representation, respectively. New features are added by adding corresponding zero values for μ_j and Σ_j for each cluster j of the new term.

VII. Experiments

We evaluated our approach based on real-world data gathered during Hurricane Sandy in 2012. We recorded tweets during the main impact of the hurricane from the 29th October 2012 till the 1st November 2012.

We considered geo-tagged tweets related to the locations Manhattan and Brooklyn. This results in 1943 tweets considering the period from 29th October 02:30 PM to 30th October 02:30 AM. Additionally, we used 676 pictures from Flickr and 366 videos related to this area. In total, we have a data set consisting of 2985 social media items for our experiments.

A. Description

We evaluated our approach compared to a purely offline feature selection method. This helps in evaluating the effectiveness of our online algorithm compared to the offline method where the optimal terms are selected before clustering. This means, we compare the following approaches:

- *Offline feature selection:* We assume that the features of each incoming document are known from the beginning. Therefore, we extracted the terms via the traditional tf-idf based on the whole considered dataset. For natural language processing and term selection, we adapted a WEKA filter [20] (i.e., the StringToWordVector Filter). It selects k-top words based on the traditional tf-idf formula.
- *Online feature selection:* As presented in Section V, where features are introduced into the clustering by considering the calculated weights.

For both feature selection methods, the 2G2M clustering algorithm [19] was used. We also tested our online approach with different settings, i.e., time and weight settings.

We extract nouns as features for the clustering, e.g., flood, Manhattan or damage, which are stemmed (using the Porter Stemmer [21]). Similar nouns are grouped together, e.g., car, automobile. This is performed by considering relations between concepts through WordNet [22]. We compared the clustering results using the *Normalized Mutual Information* (NMI) criterion [16]. The calculation of the NMI is based on the original mutual information (MI) and the entropy [16]. *The higher the NMI value, the more similar are the compared clusterings.*

Beside the comparison between the corresponding algorithms, we also evaluated the results based on their characteristics. We used the Silhouette metric [23] to identify the cluster quality. The membership grade of each known message to each created cluster is calculated. This shows how closely related messages within a cluster are. *High values of the Silhouette metric indicate well-separated clusters.*

For the online vs. offline comparison, we use the same settings. For comparison, we set for both methods the number of terms to 60. The clustering parameters are empirically evaluated and set to $\alpha = 0.01, K = 1000, \tau_\sigma = 3, T_{merge} = 8, T_{split} = 20$.

B. Results

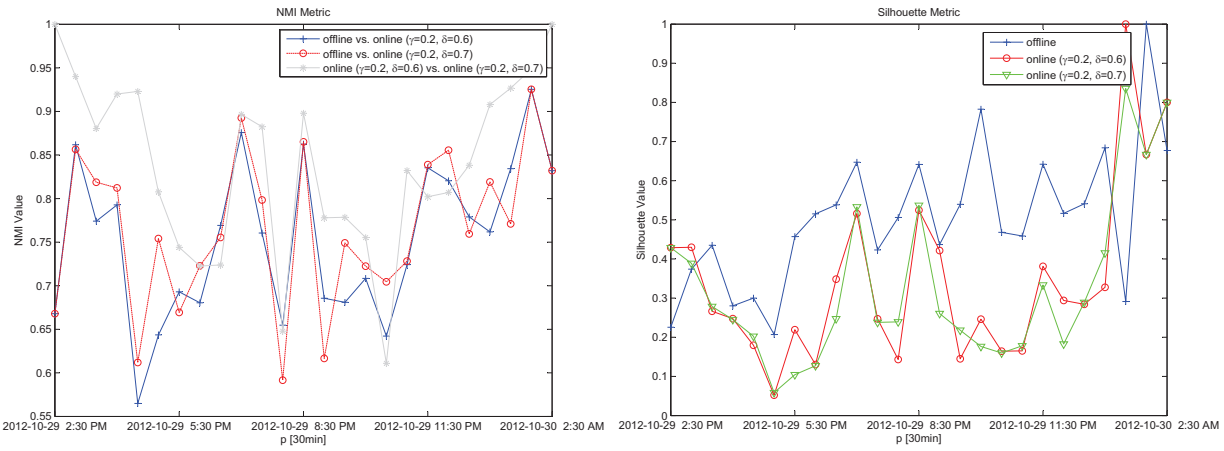
We compared the offline feature selection with our online feature selection approach. For the evaluation, we empirically evaluated and set the period to half an hour ($p = 30min$) and the sampling interval to ten minutes ($k_s = 10min$). According to the dynamics (changing topics) within the data, we set $\gamma = 0.2$ and $\delta = 0.6$ as learning and forgetting factors, respectively.

The NMI value is calculated based on the clustering results from the online and offline algorithms after each period ($p = 30min$). Fig. 2(a) shows the NMI values for each period. High NMI values indicate similar clustering results. We compared the online algorithm against the offline algorithm (see dark/solid line in Fig. 2(a)). The NMI values are high and above 0.65 except for a few periods. This indicates that the online algorithm behaves like its offline counterpart.

For evaluating the quality of the resulting clusters, we calculated the Silhouette index. Fig. 2(b) shows the Silhouette index for the settings used. It is also calculated for each algorithm after the data of each period ($p = 30min$) is processed. The dark line with the $+$ markers shows the offline algorithm whereas the other line with the o markers shows the online algorithm. The offline algorithm has a mean Silhouette value of 0.50 calculated over all periods. The online algorithm achieves a value of 0.35 with the given settings. This results in a difference of 0.16 showing a quite good performance of the online algorithm.

Beside a metric-based evaluation of the results, we also evaluated the algorithms based on a list of anecdotes (see Table I). These anecdotes describe real incidents extracted from Wikipedia that happened during Hurricane Sandy. In summary, the major incidents are flooding, power outages, and damages due to the heavy storm. Our experiments show that the online algorithm identifies most of the important sub-events. For example, it identifies sub-events related to the concept "flight" where people wrote about canceled flights. It also uncovers sub-events related to "evacuation" of hospitals. Many of the sub-events show topics on power supply and flooding. There is also a sub-event marking the flooding in the Hugh L. Carey Tunnel (see Fig. 3). There are in addition pictures showing falling trees and related damages.

The offline feature selection algorithm also indicates most of the sub-events. It creates many clusters when time passes on, which makes browsing a cumbersome task. This means that there are also clusters in a period where no item was assigned to. As long as there are unconsumed clusters, e.g., not all K clusters are fully exploited, new clusters are added. Additionally, the results of the Hurricane Sandy data show that the term set changes over time; otherwise, old clusters



(a) NMI Values (b) Silhouette Values

Figure 2. Comparison of offline and online algorithm settings

Table I
REAL INCIDENTS IDENTIFIED VIA WIKIPEDIA [24]

Major Effects	**Description (time given for 29th October, local time)**
Airports	Closed airports and canceled flights, i.e., JFK, LaGuardia, Newark (8 PM)
Tunnels	Closed tunnels (7 PM) mainly due to flooding; heavily affected: Hugh L. Carey Tunnel
Evacuation	Several hospitals and FDNY Emergency Medical Services were evacuated
Flooding	Different districts, in addition also tunnels and sub-ways
Electricity	Power outages in several districts, e.g., Manhattan, Queens, Staten Island, Brooklyn, Bronx, etc.
Fire	Several fires due to fallen trees or blown-up transformers, e.g., Breezy Point, Queens (approx. 11 PM)
Wind	Fallen trees and broken branches, damage due to the heavy wind, crane collapsed, Rockaway line affected, Statue of Liberty

would be repeatedly used during the whole time span. The online algorithm results in a lower number of clusters due to considering the changing term set. In summary, as long as people write about a specific incident, the algorithm can identify it as a sub-event.

C. Settings and Discussion

Additional experiments considered the weight and time settings. Before introducing the two formulae for describing the learning and forgetting model by separate expressions, we used a more simplified weighting mechanism (see Eq. 5).

$$g_{f,t} = (1-\gamma) \cdot u_{f,t} + \gamma \cdot g_{f,t-1} \tag{5}$$

This mechanism does not separate between learning and forgetting. Hence, if there is a peak of incoming data at the beginning of the period, it is difficult to find the right

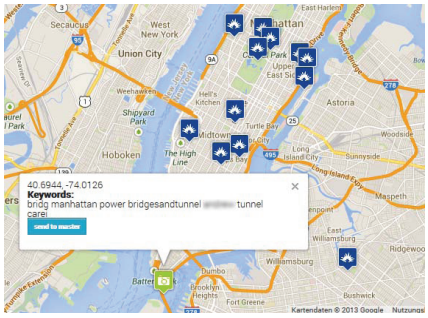


Figure 3. Sub-events of the online algorithm at 01:30 UTC (names of persons blurred, Markers by MapIcon-Collection mapicons.nicolasmollet.com)

setting for the weight to identify also sub-events originating at the beginning of the period. With the distinction between learning and forgetting (see Eq. 2), weights for the incoming data are learned much faster. Via the model in Eq. 2, the memory effect is described in a better way.

Parameter settings depend on the dynamics of the system. Hence, varying the time settings also influences the results. The tf-idf values change by varying the time period. A smaller time scale gives a deeper insight into the stream, but there must be a tradeoff between the number of messages received and the period for the clustering so that clustering makes sense.

The results could also be changed by introducing different learning and forgetting factors, to keep terms longer in memory. For example, changing the weights from $\delta = 0.6$ to $\delta = 0.7$, i.e., remembering features longer, the system also identifies the keyword *fire* in the last period, indicating information on the fire at Rockaway Park near Breezy Point (see Fig. 2(b) with triangle markers; average: 0.325). The factors influence the "memory" of the algorithm. The difference based on the NMI for both online algorithm

settings can be found in bright/light grey in Fig. 2(a), which shows slightly different assignments to clusters. Compared to the offline algorithm (see Fig. 2(a); red/dashed line), the efficiency of the online algorithm with this setting slightly increases compared to the old setting $\delta = 0.6$.

Based on the different influences of the parameters, it is important in the next step to find a suitable algorithm for adjusting parameter settings based on the concept drift.

In summary, the online algorithm compared against the offline approach identifies specific labeled clusters, e.g., power, evacuation, bridge, etc. Additionally, the online algorithm needs fewer clusters due to the changing sub-events.

For evaluation, we define parameters empirically, e.g., the number of terms, for each algorithm. In the future, we plan to change this by using an automatic approach to identify and adjust the parameters for the clustering algorithm on-the-fly. Additionally, we want to study the effects of the parameters themselves on the clustering results. We also want to extend our evaluation by labeling the data (spatial-temporal) so that item-to-cluster assignment can be evaluated in detail.

VIII. Conclusion

In this paper, we describe an approach for identifying sub-events in real time based on data gathered from social media during a crisis. It tracks the evolution of indexing terms (vocabulary) as topics (description of sub-events) continuously changing over time as new documents arrive. In particular, the approach relies on an online feature selection mechanism combined with an online clustering algorithm, 2G2M. The latter was adapted to enable clustering of an online stream of documents taking into account the change of the indexing vocabulary. The approach is evaluated using a snapshot of social media data (i.e., Twitter, Flickr, and YouTube) related to Hurricane Sandy. A comparative study of the proposed approach against its offline counterpart is presented as well. The obtained results show great potential for the use in emergency management.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°261817 and was partly performed in the Lakeside Labs research cluster at Alpen-Adria-Universität Klagenfurt.

References

[1] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness," in *Proc. of the 28th Int'l Conf. on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1079–1088.

[2] R. Nallapati, A. Feng, F. Peng, and J. Allan, "Event Threading within News Topics," in *Proc. of the 13th ACM Int'l Conf. on Information and Knowledge Management*, ser. CIKM '04. New York, NY, USA: ACM, 2004, pp. 446–453.

[3] D. Pohl, A. Bouchachia, and H. Hellwagner, "Automatic Identification of Crisis-Related Sub-events Using Clustering," in *11th Int'l Conf. on Machine Learning and Applications*, vol. 2, Dec. 2012, pp. 333 –338.

[4] M. Osborne, S. Petrović, R. McCreadie, C. Macdonald, and O. Iadh, "Bieber no more: First Story Detection using Twitter and Wikipedia," in *Proc. of SIGIR 2012 Workshop on Time-aware Information Access*, 2012.

[5] D. Chakrabarti and K. Punera, "Event Summarization Using Tweets," in *In Proc. of the Int'l AAAI Conf on Weblogs and Social Media*, 2011.

[6] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao, "Twitcident: Fighting Fire with Information from Social Web Streams," in *Proc. of the 21st Int'l Conf. companion on World Wide Web*, ser. WWW '12 Companion. New York, NY, USA: ACM, 2012, pp. 305–308.

[7] J. Rogstadius, V. Kostakos, J. Laredo, and M. Vukovic, "A Real-Time Social Media Aggregation Tool: Reflections from five Large-Scale Events," in *ECSCW 2011 CSCWSmart? Collective Intelligence and CSCW in Crisis Situations*, 2011.

[8] 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. of the 2011 Annual Conf. on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 227–236.

[9] C. Shen, F. Liu, F. Weng, and T. Li, "A Participant-based Approach for Event Summarization Using Twitter Streams," in *Proc. of NAACL-HLT*, 2013, pp. 1152–1162.

[10] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma, "An Evaluation on Feature Selection for Text Clustering," in *Proc. of the 20th Int'l Conf. on Machine Learning*, Washington DC, 2003.

[11] H. Liu and L. Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491 – 502, April 2005.

[12] F. Beil, M. Ester, and X. Xu, "Frequent Term-Based Text Clustering," in *Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 436–442.

[13] A. Bouchachia and R. Mittermeir, "Towards Incremental Fuzzy Classifiers," *Soft Comput.*, vol. 11, no. 2, pp. 193–207, 2007.

[14] X. Wu, K. Yu, H. Wang, and W. Ding, "Online Streaming Feature Selection," in *Int'l Conf. on Machine Learning*, 2010.

[15] I. Katakis, G. Tsoumakas, and I. Vlahavas, "On the Utility of Incremental Feature Selection for the Classification of Textual Data Streams," in *Advances in Informatics*, ser. Lecture Notes in Computer Science, P. Bozanis and E. Houstis, Eds. Springer Berlin Heidelberg, 2005, vol. 3746, pp. 338–348.

[16] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[17] T. Brants, F. Chen, and A. Farahat, "A System for New Event Detection," in *Int'l ACM Conf. on Research and Development in Informaion Retrieval*, ser. SIGIR '03. New York, NY, USA: ACM, 2003, pp. 330–337.

[18] P. D. Cha and J. I. Molinder, *Fundamentals of Signals and Systems: A Building Block Approach*. Cambridge University Press, 2006.

[19] A. Bouchachia and C. Vanaret, "Incremental Learning Based on Growing Gaussian Mixture Models," in *Int'l Conf. on Machine Learning and Applications and Workshops*, vol. 2, Dec. 2011, pp. 47 –52.

[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, November 2009.

[21] M. F. Porter, "An Algorithm for Suffix Stripping," *Program: Electronic Library and Information*, vol. 14, pp. 130–137, 1980.

[22] C. Fellbaum, *WordNet: An Electronic Lexical Database*, C. Fellbaum, Ed. Cambridge, MIT Press, 1998.

[23] P. J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal on Comput. and Appl. Math.*, vol. 20, pp. 53–65, 1987.

[24] Wikipedia Article. (2012, Aug.) Effects of Hurricane Sandy in New York. [Online]. Available: http://en.wikipedia.org/wiki/Effects_of_Hurricane_Sandy_in_New_-York