Enhancing Disaster Situational Awareness via Automated Summary Dissemination of Social Media Content

Shanshan Zhang, Amitangshu Pal, Krishna Kant and Slobodan Vucetic
Computer and Information Sciences, Temple University, Philadelphia, PA 19122
E-mail:{zhang.shanshan,amitangshu.pal,kkant,vucetic}@temple.edu

Abstract—The paper proposes a situational awareness service, named StayTuned that collects information from social media, extracts relevant messages, and broadcasts them to the subscribers through wireless emergency alert system. StayTuned uses automated filtering and summarization of messages and updates subscribers with real-time situational summaries. Extensive experiments were conducted using twitter data collected during the Sandy hurricane to evaluate performance of the automated message extraction.

Index Terms—Emergency update application, twitter, crisis management, situational awareness, filtering, summarization.

I. INTRODUCTION

During a natural or man-made emergency or a disaster, information useful for situational awareness typically arrives from multiple sources, including content generated by people within the impacted areas and data collected by emergency responders. As an emergency unfolds, it is likely that communication infrastructure becomes damaged and heavily strained and that the demand for sending and receiving information vastly exceeds the available capacity. One of the major challenges in disaster response is maintaining flow of information, such that valuable information generated within a disaster area could be disseminated and that the impacted population could stay informed. This paper proposes a system aimed at maintaining flow of information during emergencies and disasters.

While there are multiple information sources, in this paper we focus on collection and dissemination of Twitter messages generated during a disaster. As evidenced by several past disasters, Twitter has established itself as the disaster communication vehicle of choice due to its modest networking requirements, ease of use, and brevity. For example, soon after the 2011 Japanese earthquake, a volume of disaster related tweets exceeded 5,500 per second. Twitter has also been instrumental during a wide variety of emergencies and disasters, such as fires, floods in 2009 [2], and terrorist attacks. Although there have been many recent studies on the use of social networks for disaster response, a uniqueness of our work is its focus on resource-aware dissemination of information. In this direction, we propose a situational awareness service named StayTuned that efficiently processes disaster related information and disseminates it in a compressed form to the impacted population. A user interface of the StayTuned is shown in Figure 1

Although it is easy to envision a fully automated system for processing and dissemination of tweets and real-time optimization of the available communication resources, such a vision is still difficult to realize using the current state of the art. Therefore, we consider a human-in-the-loop system. We assume that a large representative subset of disaster related tweets is available to the emergency authorities who could then select a subset of tweets to be broadcast to the impacted population. The cellular subscribers receive the selected messages using StayTuned. Important and representative tweets are selected in two phases. In the first phase, the disaster-related messages are selected by a classification algorithm trained using labeled and unlabeled data from the current and previous related emergencies. Whenever a large group of people are reporting about events within a limited area, it is to be expected that generated messages have significant level of redundancy. To reduce redundancy, during the summarization phase a small subset of non-redundant and informative messages are selected by a linear programming algorithm. Extensive simulations using real tweets show that such filtering and summarization process can extract a small fixed subset of messages which are highly representative of the generated stream of disaster-related messages.

The outline of this paper is as follows. Section II proposes the network architecture and protocols for StayTuned system implementation. Section III introduces the proposed filtering and summarization framework for extracting a representative subset of relevant messages from the available message pool. Section IV describes the experimental setup and shows results obtained by using the tweets collected during the Sandy Hurricane. Discussion of related work is provided in section V. Finally, the conclusion is provided in section VI.

II. NETWORK ARCHITECTURE FOR StayTuned

Public Warning System (PWS): In an event of an emergency, the PWS authorities need to provide warnings and update information to the people in an efficient manner. To quickly and effectively alert and warn the public about serious emergencies, Wireless Emergency Alerts (WEAs) is made available through the Integrated Public Alert and Warning System (IPAWS) infrastructure. All major cellular providers

Figure 1: StayTuned UI
The implementation of a CBS, illustrated in Figure 2, consists of

LTE standards [3]. CBS is also included in current 3GPP 2G, 3G standards. We choose CBS for our StayTuned service as illustrated in Figure 3. The first component is a trainable filter which removes irrelevant messages by using a model trained from the current and previous similar disasters. The second component of the message extraction framework is an adjustable summarizer, which selects representative messages pertinent to the disaster while removing the redundancies.

Network architecture for disseminating CBS: The basic implementation of a CBS, illustrated in Figure 2 consists of cell-broadcast centers (CBCs) and at least one cell-broadcast entity (CBE). CBE resides under a trustworthy or government authority domain. CBE is the message creator that compiles the messages, specifies the locations where they need to be broadcasted and passes them to the CBC operators. The operators then send the cell broadcast messages to the respective targeted areas.

In majority of cases, the CBEs are further divided into two units. The first is a message generation unit that accumulates emergency information from different sources and generates messages that are to be sent. These messages are then sent to the authentication gateway unit that authenticates the sender and validates legitimacy of the messages before sending them to the CBC service providers. To ensure the security and integrity of the messages, the payloads may also be encrypted [3].

In our StayTuned system, the messages are collected from different sources, e.g., from different social media. During disasters, the number of collected messages can be overwhelming and redundant. A viable option is to filter and extract a subset of relevant messages from the incoming message stream before forwarding them to the CBCs.

III. StayTuned Message Extraction Framework

We propose a two-step message extraction framework for StayTuned service as illustrated in Figure 3. The first component is a trainable filter which removes irrelevant messages by using a model trained from the current and previous similar disasters. The second component of the message extraction framework is an adjustable summarizer, which selects representative messages pertinent to the disaster while removing the redundancies.

Figure 2: An illustration of a CBS PWS architecture

Figure 3: Message extraction framework.

A. Message filtering

During disasters, only disaster-related messages are of interest for StayTuned. The goal of message filtering is to discard less relevant messages while keeping the important ones. We formulate the message filtering problem as a text classification task. The problem setup is explained as follows.

Let us assume there is a set of p labeled messages $M_{train} = \{(m_1, y_1), (m_2, y_2), ..., (m_p, y_p)\}$, where each message $m_i$ is converted into a feature vector $x_i \in \mathbb{R}^n$, and $y_i$ is a binary label indicating whether the message is disaster-related. The objective is to train a model $y = f(x)$ whose output can be used to classify messages. We will assume that the output is real-valued and that larger values indicate a stronger likelihood of being disaster-related.
To learn an accurate filter, a large number of labeled messages is preferred. However, during the emerging disaster, the number of labeled messages is likely very small. In our previous work [5], we addressed the small labeled training set problem with the help of a large amount of unlabeled messages from previous disasters, \( \mathcal{M}_{\text{ unl}} = \{m_1, m_2, \ldots, m_q\} \), where \( q \gg p \), in which labels \( y_i \) are unavailable. We used word2vec [6] algorithm on unlabeled messages to learn how to represent each word as a real valued vector in \( \mathbb{R}^d \). word2vec is able to represent related words as similar vectors. Given the vector representations, we then used k-means algorithm to group words into \( k \) clusters. We list words from several disaster-related clusters in Table I as an illustration. Finally, we represented each message as a bag-of-clusters, which is a count vector of length \( k \). The \( i \)-th element of the bag-of-clusters equals the number of words from the message that belong to the \( i \)-th cluster. We observed that such representation of messages allows training of accurate classifiers even from very small training sets with a few dozen of labeled messages.

### Table I

**CRISIS-RELATED CONCEPTS LEARNED.**

<table>
<thead>
<tr>
<th>Concept</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flooding</td>
<td>weather, storm, flood, storms, flooding, floods, thunderstorms, #storm, #floods, #flooding</td>
</tr>
<tr>
<td>Power</td>
<td>#ndl, #bier, #stages, #pk, #hb, #sticktogether, #newnorma, #droughts, #powerout</td>
</tr>
<tr>
<td>Time</td>
<td>day, time, night, weekend, minutes, min, mins, wknd, offseason, midday, arvo, midnite, w/end, wend,</td>
</tr>
<tr>
<td>Verbs</td>
<td>thinking, waiting, telling, hoping, wishing, praying, searching, seeking, needing, demanding</td>
</tr>
</tbody>
</table>

### B. Message Summarization

The disaster related messages are likely to be redundant. For example, it is common that multiple people report on important events such as flooding or food shortage. Thus there is an opportunity to further reduce the number of messages by summarization. Let us suppose we are given a set of \( N \) messages. We formulate the problem as a subset selection problem following work in [7], [8], where the objective is to select a subset of \( n \) messages that maximize the informativeness. In this work, we also maximize semantic completeness, to be defined later in the text.

**Informativeness score:** Content words can be provided by domain experts, describing disaster situations. In our experiments, we simply use nouns, proper nouns, verbs and numbers as content words. Let us denote the set of \( V \) unique content words as \( \mathcal{CW} = \{cw_1, cw_2, \ldots, cw_V\} \). Emergency responders at CBE can define a weight for each content word \( cw_j \) as \( W(cw_j) \). Then, informativeness of message \( i \) can be calculated as the sum of the weights of its content words: \( I(m_i) = \sum_j W(cw_j) \).

**Semantic completeness:** While informativeness score is a useful objective criterion for summarization, it is not sufficient to reduce redundancy in messages. For example, suppose the following three messages are given the same informativeness score:

(i) Water has entered my house.

(ii) Streets are flooded and not one rain drop came down so far.

(iii) Attention everyone I lost power.

Instead of picking (i) and (ii) out of them, we should select (ii) and (iii), since they refer to two semantic groups and cover two aspects of disaster.

Thus, we propose a preprocessing step that clusters messages. We first represent the \( N \) messages as a matrix \( R^{N \times V} \), with value at cell \( (i, j) \) being the count of word \( j \) in message \( i \). Since the count matrix is sparse, we use the SVD algorithm to reduce the number of columns to \( d, d \ll V \). Then, we apply k-means algorithm to group all messages into \( K \) clusters.

The resulting message selection optimization algorithm is described in equation (1). The notation is explained in Table II.

\[
\text{Maximize } x_i y_j \sum_{i=1}^{N} I(m_i)x_i + \sum_{j=1}^{V} W(cw_j)y_j \\
\text{subject to } \sum_{i=1}^{N} x_i < L \sum_{i \in T(cw_j)} x_i \geq y_j \sum_{j \in W(m_i)} y_j \geq |W(m_i)|x_i \sum_{i \in C_k} x_i \leq \tau \sum_{i=1}^{N} x_i, \text{ for } k \text{ in } 1, 2, \ldots, K.
\]

The above optimization maximizes the informativeness and the semantic coverage of the selected messages. The first constraint specifies the budget, which is the number of messages the summarizer wants to extract. The second and third constraints ensure validity of the selected messages, i.e., all of the content words in a selected message are selected and a selected content word must reside in a selected message. The final set of \( K \) constraints defines a maximum number of messages that could be selected from any given cluster, which is controlled by parameter \( \tau \in (0, 1] \). The formulated problem is solved using GUROBI Optimizer [9].

### Table II

**NOTATION USED IN THE SUMMARIZATION STAGE.**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_i, y_j )</td>
<td>Binary values with 1 if the message ( i ) or the content word ( j ) is selected and 0 otherwise</td>
</tr>
<tr>
<td>( I(m_i) )</td>
<td>Informativeness score of message ( i )</td>
</tr>
<tr>
<td>( W(cw_j) )</td>
<td>Weight of content word ( j )</td>
</tr>
<tr>
<td>( L )</td>
<td>Maximum number of messages can be selected</td>
</tr>
<tr>
<td>( T(cw_j) )</td>
<td>Indexes of messages of which ( cw_j ) is a member</td>
</tr>
<tr>
<td>( W(m_i) )</td>
<td>Indexes of words in message ( m_i )</td>
</tr>
<tr>
<td>( C_k )</td>
<td>Indexes of messages in semantic cluster ( k )</td>
</tr>
<tr>
<td>( K )</td>
<td>Total number of semantic clusters</td>
</tr>
<tr>
<td>( N )</td>
<td>Total number of messages in the pool</td>
</tr>
</tbody>
</table>
IV. EXPERIMENTAL SETUP

A. Data Sets

We evaluated performance of filtering and summarization in StayTuned using real world data. We collected 4.7 million tweets posted during 2012 Sandy Hurricane, which affected the U.S. Northeast from 10/28/2012 to 11/03/2012. We focused on a subset of tweets generated from 4 large metro areas in the region: New York, NY (NYC); Philadelphia, PA (Phila.); Boston, MA (Boston) and Washington, DC (DC).

B. Simulated Settings

Typically, a CBE will divide the targeted area into small cells (e.g., school districts). Since we did not have this information handy, we divided each of the 4 cities into a grid of equally sized cells with size $D_{fix} \times D_{fix}$. Note that in this setting the traffic load of each cell is highly imbalanced, because of varying population densities in typical big cities. Figure 4 shows the approximate population density of New York City metro area by considering the number of unique users who tweeted. We assume $D_{fix} = 500$ for Figure 4. The dense regions are denoted by dark color. The figure shows that the population is highly concentrated on the island of Manhattan.

Figure 4: The Manhattan area is partitioned into cells with size $D_{fix} = 500m$. The tweet densities are shown in different color scale.

Dividing a targeted area into small regions results in a large number of cells. For example, since New York region is of size $48km \times 48km$, we get around 10,000 cells of size $500m \times 500m$. We discovered that the densest 10% of cells cover ~70% of the population. The long tail effect is illustrated in Figure 5. Since for the less dense cells the network congestion is not a significant issue, we only evaluated our filtering and summarization algorithms on the densest 10% of the cells.

Filter: To train the filter we used a labeled dataset $M_{train}$, obtained from [10]. It contains 3,505 tweets labeled as 1 if they are related to Sandy Hurricane and 0 otherwise. To create the word clusters we downloaded word vectors pre-trained by wort2vec algorithm on 400 million unlabeled tweets from the Internet. Then, we created 2,000 clusters using k-means algorithm. We then trained the filter using logistic regression.

<table>
<thead>
<tr>
<th>NYC</th>
<th>Phila.</th>
<th>Boston</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>M</td>
<td></td>
<td>444,016</td>
</tr>
<tr>
<td>$</td>
<td>S</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

TABLE III: NUMBER (PERCENTAGE %) OF MESSAGES AFTER FILTERING.

Summarizer: The key to the summarizer is the definition of the content words and their weights. In reality, the content words and their weights can be passed from authorities or domain experts to better guide the summarization. For simplicity, we extract content words automatically from all tweets in an area as follows:

1) Content words: We first use the tweet POS (part-of-speech) tagger from CMU [11] to tag all of the tweets from a city. Then nouns, proper nouns, numbers and verbs with more than 5 appearances are selected as content words.

2) Weights of content words: For our experiments, we scored content word $cw$ as: $W(cw) = \log_2(f_{cw} + 1) \times \frac{N}{T(cw)}$, where $f_{cw}$ is the total count of $cw$ and $T$ is the number of tweets containing $cw$.

Another important decision for the summarizer is the message clustering procedure, which helps us achieve a better variety of selected messages. In our experiment we clustered messages in a cell into 10 clusters and removed the smallest 2 clusters.

V. EXPERIMENTAL RESULTS

We evaluated the performance of our proposed message extraction framework for StayTuned both quantitatively and
Caution and advice

Displaced people and evacuations

Donation needs/volunteering services

Infrastructure and utilities damage

Missing, trapped, or found people

Sympathy and emotional support

One standard and widely accepted crisis-related ontology proposed in [12], [13] contains 8 categories (C₁–C₈) as shown in Table V. We measured the distribution of message categories of the filtered tweets and the final summarizations. To get the number of tweets in each category, we took a labeled data set provided by [14], which contains 2,013 tweets from Odile Hurricane in 2014, and trained a multi-class classifier using the described method for the filter component. Then the classifier was used to predict the category of tweets from Sandy Hurricane.

We first get the set of tweets after filtering S₁ for each city from Table III. We also get a set of tweets after summarization S₂ for each city using L = 50, Dfix = 2000m and τ = 0.125 from Table IV. Therefore, the percentage of category k in S₁ is defined as: p_S₁(k) = n_k / |S₁| × 100%, i ∈ {1, 2}, where n_k is the number of tweets in S₁ falling in category k. Table VI shows how the distribution of message categories changes from S₁ to S₂.

One observation is that the eight categories are not evenly distributed. A small proportion of tweets is classified into categories 1, 2, 3 and 6. It might be because the death or injury of people were rare events. Fair amount of tweets was related to infrastructure damage, donations, and words of sympathy. Many tweets were classified as other useful information.

Comparing the distribution of message categories in S₁ and that in S₂, we do not see dramatic differences. However, the percentage of tweets of category 8 is reduced by 5% – 10%, while the percentages of other categories are lifted slightly. The results illustrate that the summarizer reduced the imbalance slightly.

Finally, we set L = 5, Dfix = 1000m, τ = 1 (by which we do not force the clustering constraints) for all cities and apply our message extraction framework on the most dense cell in each city. The most important 5 messages from the 4 cities are shown in Table VII. The two numbers in the parentheses denote the number of selected tweets and the total number of tweets, respectively. During Sandy Hurricane, the Manhattan area was considerably affected. Therefore, damage and disruptions, such as flight delay, power outage, were commonly reported. Users from the other three cities were worried about power outage and were sending prayers to the people along the eastern shore, but there is much less reporting qualitatively.

### Table V

Crisis Related categories [12], [13].

<table>
<thead>
<tr>
<th>Categories</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₁</td>
<td>Injured or dead people</td>
</tr>
<tr>
<td>C₂</td>
<td>Missing, trapped, or found people</td>
</tr>
<tr>
<td>C₃</td>
<td>Displaced people and evacuations</td>
</tr>
<tr>
<td>C₄</td>
<td>Infrastructure and utilities damage</td>
</tr>
<tr>
<td>C₅</td>
<td>Donation needs/volunteering services</td>
</tr>
<tr>
<td>C₆</td>
<td>Caution and advice</td>
</tr>
<tr>
<td>C₇</td>
<td>Sympathy and emotional support</td>
</tr>
<tr>
<td>C₈</td>
<td>Other useful information</td>
</tr>
</tbody>
</table>

We would like to check the percentage of messages kept after filtering and after summarization. Suppose M is the set of all tweets for a specific city and S₁, S₂ are the remaining tweets after filtering and after summarization respectively, the percentage of messages kept in Sᵢ is then defined as: p(Sᵢ) = |Sᵢ| / |Sᵢ| × 100%, i ∈ {1, 2}.

The numbers (percentages) of messages kept after filter are reported in Table III. The fraction of crisis-related tweets is twice larger in NYC compared to other cities, which is expected because NYC was more heavily impacted by Sandy Hurricane.

Then, we checked whether our summarizer works as expected when we vary the cell size Dfix and the message budget L. We explored the following limits on the number of messages per cell, L = [10, 20, 50, 100]. The cell size Dfix ranged from [500m, 1000m, 2000m, 5000m]. We fixed τ = 0.125, which encourages about the same number of messages to be selected from each cluster. The numbers and percentages are reported in Table IV. From Table IV we can observe that:

1) As we increase the cell size, we will keep a smaller fraction of messages.
2) As we increase the size of budget, we will keep more messages.

### Table IV

Number (percentage %) of messages kept after summarization.

<table>
<thead>
<tr>
<th></th>
<th>Dfix = 500m</th>
<th>Dfix = 1000m</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L = 10</td>
<td>L = 20</td>
</tr>
<tr>
<td>NYC</td>
<td>2990(0.67)</td>
<td>5277(1.19)</td>
</tr>
<tr>
<td>Phila.</td>
<td>689(0.57)</td>
<td>753(0.62)</td>
</tr>
<tr>
<td>Boston</td>
<td>392(0.35)</td>
<td>441(0.31)</td>
</tr>
<tr>
<td>DC</td>
<td>466(0.46)</td>
<td>510(0.51)</td>
</tr>
<tr>
<td></td>
<td>Dfix = 2000m</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L = 10</td>
<td>L = 20</td>
</tr>
<tr>
<td>NYC</td>
<td>370(0.08)</td>
<td>740(0.17)</td>
</tr>
<tr>
<td>Phila.</td>
<td>128(0.11)</td>
<td>232(0.19)</td>
</tr>
<tr>
<td>Boston</td>
<td>80(0.09)</td>
<td>152(0.18)</td>
</tr>
<tr>
<td>DC</td>
<td>80(0.08)</td>
<td>152(0.15)</td>
</tr>
</tbody>
</table>
of the damage and disturbances.

### VI. RELATED WORK

#### Smartphone for emergency applications

The increasing penetration of smartphones now-a-days makes them a platform for various emergency applications such as health monitoring, mobile sensing, or building ad-hoc networks in crisis scenarios. Several network testbeds and protocols have been developed for search and rescue in disasters, including VigilNet [15], AlarmNet [16], Code Blue [17], DieselNet [18], and DistressNet [19]. TeamPhone [20] uses a self-rescue system for the trapped victims during disaster using WiFi-Direct communication. E-DARWIN [21], [22] proposes a WiFi-Tethering based disaster recovery network formation using wireless smartphones that are largely available in the disaster-hit areas. Argus [23] builds a 3D mapping of disaster scene using cooperative data collection from bystanders/drones present around the targeted zone to facilitate rescue operations. Unlike others, the proposed StayTuned provides a service for distributing filtered and summarized updates available from different information sources to the subscribers. This unique aspect of this paper is integrate the use of existing cellular infrastructure and analytics of big data obtained from various social media to create an agile emergency update service that is expected to provide significant additional value in future emergencies and disasters.

#### Twitter for situational awareness

Recent years have seen an increased interest by research community in studying how to exploit Twitter data for situational awareness in the emergency and disaster contexts [24], [25]. Event detection is arguably the most active research topic, where the objective is to detect new events from a real-time twitter stream. A typical approach for event detection is to define one or a few keywords (e.g., earthquake) of interest and to track online if there are temporal bursts of the keywords’ use in tweets [26]. Extensions of this approach include general-purpose detection systems that track a large number of keywords [27], phrases [28] or emergence of clusters of similar tweets [29]. More recently, researchers started paying more attention to the spatial aspect of events [30]. For example, [31] considers burstiness of term “earthquake” in both time and space to detect spatial clusters of tweets with that word as candidates for the earthquake event. The unsupervised approach for event detection can be further enhanced by adding a classifier that is trained on previous events to recognize which bursty clusters are events and which are not [32].

Summarizing and visualization of disaster-related tweets help human responders to quickly grasp the vast amounts of generated information. Representative systems are Senseplace2 [33], a visual analytics system that allows an operator to enter a query (in a form of a term or a hashtag), look at the map to observe where is the keyword common, click on a specific location, and view individual ranked tweets of the damage.
from the selected location, and Twitinfo [34], a tool that allows an operator to browse a large collection of tweets using a timeline-based display, drill down to subevents, and explore via geolocation, sentiment, and popular URLs.

VII. CONCLUSION

In this paper, we envision situation awareness of mobile service via automated data collection from the smart phone users and human directed communication such as phone calls and social media. Acknowledging that Twitter has established itself as the premier human communication mechanism during disasters and the wealth of publicly accessible disaster-related twitter data, we considered integration of only the twitter-based information for the purposes of situational awareness. The key objective was to understand how to utilize twitter data for updating the subscribers regarding the important updates by leveraging the previous works on event detection and situational awareness from Twitter data. As such, our proposed work automatically selects, orders and summarizes Twitter data at a large scale, which can be deployed in real-time during disasters via the message broadcasting service StayTuned.

ACKNOWLEDGMENTS

This research was supported by JST-NSF joint funding, Strategic International Collaborative Research Program, SICORP on Japan side, and CNS-14461932 award from NSF.

REFERENCES


