Social Fusion: Integrating Twitter and Instagram for Event Monitoring

Prasanna Giridhar, Shiguang Wang, Tarek Abdelzaher, Tanvir Al Amin
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, Illinois - 61801, USA

Lance Kaplan
US Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783, USA

***Abstract*—This paper describes the implementation of a service to identify and geo-locate real world events that may be present as social activity signals in two different social networks. Specifically, we focus on content shared by users on Twitter and Instagram in order to design a system capable of fusing data across multiple networks. Past work has demonstrated that it is indeed possible to detect physical events using various social network platforms. However, many of these signals need corroboration in order to handle events that lack proper support within a single network. We leverage this insight to design an unsupervised approach that can correlate event signals across multiple social networks. Our algorithm can detect events and identify the location of the event occurrence. We evaluate our algorithm using both simulations and real world datasets collected using Twitter and Instagram. The results indicate that our algorithm significantly improves false positive elimination and attains high precision compared to baseline methods on real world datasets.**

I. Introduction

The main contribution of this paper lies in developing a service that uses a fusion algorithm for physical event detection from *multiple social networks* as a way to improve the accuracy of event detection. Specifically, we fuse data feeds from Twitter [4] and Instagram [2]. The two networks have complementary advantages. Twitter data are more prolific (500 million tweets posted per day [5]), leading to detection of more events, but as shown in our evaluation, it is also more noisy, generating more false-positives. In contrast, Instagram data feeds are sparser (80 million images posted per day [3]), which leads to the benefit of fewer false positives at the expense of detecting fewer events. We show that fusing the two together can offer a solution that features the benefits of both; the results have a much smaller fraction of false positives compared to using Twitter alone, and have more events detected, compared to Instagram. We believe that the solution described in this paper offers a new point in the trade-off space between precision and recall in event detection techniques from social media data, aiming to combine the benefits of past solutions.

The key underlying analytical contribution lies in a new expectation maximization algorithm that enables even detection using fusion of data feeds from different social networks. By combining data from multiple social media, we are able to detect events that may not have enough corroboration in one network or be indistinguishable from "noise" in another. The algorithm considers the smaller of the data feeds (presently, it is Instagram). For each object in that feed, it attempts to find related objects in the larger feed (Twitter). It then uses a novel model to statistically estimate the likelihood that the found set of data objects describe a consistent event. If so, an event is said to have been detected. Events detected using this algorithm strike a better balance between false positives and false negatives, compared to either network in isolation, which is the main contribution of the new work.

The paper builds on a long history of event detection from social media. Among the first efforts in that context is the work on earthquake detection from Twitter [19] where each user is considered as a sensor node that reports a target event according to some probabilistic distribution. Since then many other works [25], [15], [28], [27] have exploited statistical properties of tweets to identify ongoing events. A second popular social network, Instagram, allows users to share pictures of their observations. The idea of event detection from Instagram dates back several years [18], [22], [20], [21]. Unlike Twitter, where only 1.5% of the Twitter data is geo-tagged [17], Instagram has a significantly higher fraction (15%) of data with location information [12]. In our own previous work, we described a system that uses feeds from Twitter [11] (alone) and Instagram [12] (alone) to detect events. This paper builds on such prior work by offering a novel fusion algorithm that aims to offer a better trade-off between precision and recall of the individual approaches.

The rest of this paper is organized as follows. Section II describes the complete architecture of the system we implemented on which the fusion algorithm runs. In Section III we present the problem formulation and the algorithm of our approach. The evaluation is discussed in Section IV. Related work is described in Section V. Finally, conclusions are presented in Section VI.

II. Service Architecture

Our service consists of several runtime modules as illustrated in Figure 1. The functionality of each module is described below:

1) *Crawler*: This module is provided with a *Task Info* file which contains the keywords entered by the user to initiate the search query. We crawl data every one hour from both Twitter and Instagram with the help of APIs and store it on the disk by grouping data into daily bins.
2) *Tweet Clustering Module*: We use the clustering approach described in [6] to remove all the redundant

2474-0756/17 $31.00 © 2017 IEEE
DOI 10.1109/ICAC.2017.46

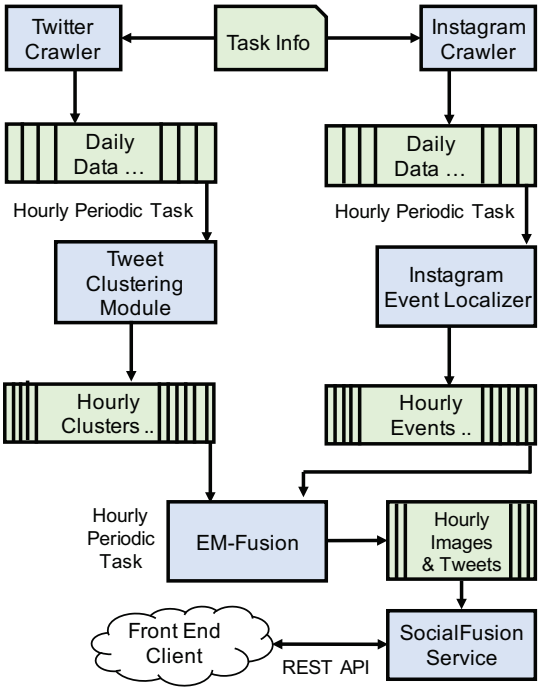


Fig. 1: Implementing Social Fusion as a Service

tweets. This module is initiated as soon as a bin gets updated by the crawler module and outputs a set of hourly clusters in a separate file on the disk.

3) *Instagram Event Localizer*: This module runs an user provided event detection technique on the crawled data from Instagram and generates hourly event clusters for each bin.
4) *EM-Fusion*: This module contains the main fusion algorithm which we describe later in the paper. It reads the data from disk generated by previous two modules and produces an output for all detected events comprising of tweets and images.
5) *Social Fusion Service*: The last module of our service takes care of interaction with the front end client and the output generated using the fusion algorithm.

III. Problem Formulation and Solution Approach

In this section, we first provide an example of (manual) fusion using data collected from Twitter and Instagram. We then formulate the automated data fusion problem and derive an algorithm to detect events from both social networks.

A. A Manual Fusion Example

In our previous work, we used Instagram and Twitter separately as social networks to localize events in urban spaces. In order to understand whether fusion is feasible, we need to establish that the same events can leave a signature on both networks. Towards that end, we collected data on the topic of protests (i.e., collected tweets and Instagram images tagged "protest"). We then clustered Instagram posts on the topic with the expectation that clusters of images containing the "protest" tag, that originate roughly from the same time and space, likely describe a protest at the indicated time and location. We conducted a small study on a few such clusters of Instagram objects, to check if the corresponding events are also mentioned on Twitter. Table I contains two examples from the Instagram *protest* dataset. Each of these correspond to cluster locations originating multiple pictures tagged as a protest. The set of all hashtags of these pictures is indicated. We scanned the Twitter *protest* dataset for the same 24 hour interval during which events were identified using our Twitter-based event detection technique [11]. The technique identifies events together with their salient keywords. It is clearly evident from the corresponding tweets and keywords that they refer to the same events and locations. Thus, we can see that a mapping exists between events detected on the two individual networks. The next step is to figure out a way that can automatically identify this mapping without user interference, even for events that are not independently detected in one or both of the networks.

The mapping between related Instagram and Twitter feeds, referred to above, is done on two steps. First, we start with the smaller feed (Instagram). For each object posted in this feed, we identify all *potentially related* posts in the larger (Twitter) feed. Second, with the set of potentially related posts identified, we make a decision on whether they correspond to a real event, or whether the similarity is accidental. These steps are described in the next subsections, respectively.

B. Finding Potentially Related Posts

To find which Instagram posts are potentially related to which Twitter posts, we need a logical distance metric between an Instagram post and Twitter posts. A convenient metric is the location referred to in the post. However, most tweets do not mention location. Thus, we also need to consider keywords. Instagram posts contain image tags (*hashtags*). We therefore need to identify whether words contained in a tweet are related to these hashtags or not. In this paper, we choose a "quick and simple" approach that relies on string matching, but does not consider semantics. Better algorithms can be developed by considering semantic distance between different strings.

In developing a string-matching approach, an important question is which string to match? A further look at the event examples in Table I reveals that not all the Instagram hashtags are equally important in finding matching tweets. For example, in case of the first event "*westboro*" and "*westborobaptistchurch*" are the only significant tags that help identify the three tweets corroborating the detection. Similarly, in the case of the second event "*googlebus*", "*gentrification*", "*valenciacorridor*", and "*displacement*" are the significant tags that can identify the related tweets. We need to define a metric that helps us find all the tweets that are potentially related to a given set of Instagram hashtags.

To reduce the noise, we first do some preprocessing on tweet text by removing the english stopwords, special characters (non alphanumeric), and weblinks. We also do not consider the query keyword (e.g., *protest*) as it will be present in all the Instagram/Twitter posts by default.[1] It is also

[1]Remember that in our example, the data was collected by querying Instagram and Twitter for all posts containing the query word, "protest".

TABLE I: Event match examples using Instagram and Twitter

Instagram Location	Instagram Tags	Tweets	Event Signature
(39.045417, -95.721562)	['picket', 'brainwashed', 'westboro', 'protest', 'important', 'wbc', 'truth', 'spreadtheword', 'westborobaptistchurch', 'true', 'dontworrybehappy']	(1) you realize christians protest westboro baptists right is wrong (2) westboro baptist church really protest gunderson production laramie project put years ago (3) fisher westboro protest offers gunderson students opportunity show grizzly pride	(westboro, protest)
(37.7870288, -122.407553)	['protest', 'themission', 'gentrification', 'valenciacorridor', 'googlebus', 'displacement']	(1) laylamrazavi el desalojo ya basta protest googlebus displacement gentrification valenciacorridor (2) video tech workers displaced googlebus protest catch another bus (3) tech buses blocked 45 minutes 2 yrs amp 2 months 1st googlebus protest sfbos sfmayor sb50	(googlebus, protest)

important to note that the hashtags are sometimes composed of multiple words merged together. For example, consider the first event again from Table I in which the significant tag "*westborobaptistchurch*" is actually composed of three different words - *westboro*, *baptist*, and *church*. In order to overcome this issue, we use the processed tweet text and remove all the white spaces to form a single string. Next, we determine the number of hashtags from the Instagram post that are present as substring within the modified tweet string. This metric known as *tag similarity* is defined as below:

$$tag\_sim = \frac{\textit{\# of tags present as substring in tweet string}}{\textit{\# of tags}} \tag{1}$$

Based on equation 1, the similarity score for the tweet - "*westboro baptist church really protest gunderson production laramie project put years ago*" will be $\frac{2}{10}$ (We do not consider the query keyword (*protest*) which was used to collect the datasets in this calculation). Thus the only tags that are present as substrings within the main string are "*westboro*" and "*westborobaptistchurch*".

TABLE II: Top 5 tweets for Instagram location using tag similarity metric

Instagram Location: (-33.89102, 151.277726) **Location Name**: Bondi Beach, New South Wales, Australia **Tag Similarity Tweets**
(1) great symbolic protest happening right bondi beach sydney bondi electorate turnbull time (2) the people wentworth tell letthemstay protest morning bondi beach (3) photos morning letthemstay protest bondi (4) people gather across australia protest return asylumseekers naru letthemstay (5) saudiarabia wants behead teenager taking part protest humanrights humanity bbc

Using the above defined metric we can now identify tweets that are potentially related to a given Instagram post. If multiple Instagram posts originate from the same location, we can combine their tags and compute distance of individual tweets with respect to that combined tag set. This distance will yield similarity to a potential event at the given location. Table II shows the top five tweets using the metric for a given Instagram location. We emphasize that these are *potentially relevant* tweets. We do not yet know, based on the above distance metric alone, if they are truly relevant or not (i.e., only accidentally similar). A contribution of our work, described below, is to offer a maximum likelihood estimate of actual relevance. This algorithm leads to the discovery of three separate quantities: (i) whether an Instagram location is an actual event location or not, (ii) for a given Instagram event location, what are the significant tags and the corresponding relevant tweets corroborating the observation, and (iii) what is the exact geo-coordinate where the event happened. We propose an unsupervised method in which we assume that we have no prior knowledge of the significance of the Instagram tags as well as the relevance of the tweets using the above similarity metric. The details of this model are described in the following subsection.

C. Fusion Model

Let us assume that a selected Instagram event detection technique generates cluster $(E_1.E_2....E_K)$ within a time interval. We then identify the union of the hashtag words $W_1, W_2, ...W_M$ that are present in each event cluster E_k. With the help of the geo-tagged coordinates associated with a cluster we also retrieve the exact location name using the Google Maps API [1] service. This location name is of the form $L_1, L_2, ...L_L$ where each L_l is a component in the address hierarchy L. Let T be the set of tweets $T_1, T_2, ...T_N$ retrieved using the tag similarity metric for the hashtags. Since a tweet can have more than one hashtag, we define A_i as the signature (comprising of one or more hashtags) which retrieves the tweet T_j. We define R_j as the relevance variable $(R_j \in \{0, 1\})$ indicating if a particular tweet T_j is relevant to an event cluster E_k or not. For every hashtag signature A_i we have a group of associated tweets. This enables us to find the average word vector that can be related to the hashtag signature A_i. We define the average word vector as the list of all distinct words from the associated tweets using their average count. We also link L_l to a tweet T_j depending on whether the location name appears in the tweet or not. The definition of all the notations used are mentioned in Table III.

For every tweet T_j we can now define a score based on its distance (using cosine similarity) from the average word vector of corresponding hashtag signature A_i. It can be assumed that all the relevant tweets are more likely to represent the same information. Thus a hashtag signature A_i generating relevant tweets will have an average word vector close to all the relevant tweets resulting in high similarity scores. Whereas a tag signature generating noisy tweets will produce a word vector that results in low similarity scores. We define this

TABLE III: Definition of Notations

E_k	Instagram Event Cluster
A_i	Signature composed of hashtags used in a cluster E_k
T_j	Tweet associated with a cluster E_k
L_l	Location name associated with a cluster E_k
R_j	Relevance of a tweet $\in \{0, 1\}$
C_{ij}	Coherence score using word vector of hashtag A_i and corresponding tweet T_j
L_{lj}	Indicator if location L_l appears in $T_j \in \{0, 1\}$
$B(\alpha, \beta)$	Beta distribution with parameters α and β

property as *Coherence* which tries to distinguish between the two set of classes. At the same time we also use the location information to increase the confidence of our assumption. For every location name L_l we define p_l as the probability that it appears in the tweet T_j given that it is relevant and q_l as the probability that it appears in the tweet T_j given that it is not relevant. Mathematically, we can define these terms as follows:

$$p_l = P(L_{lj} = 1 | R_j = 1) \tag{2a}$$
$$q_l = P(L_{lj} = 1 | R_j = 0) \tag{2b}$$

We consider that a location name is more likely to be a part of relevant tweet than the irrelevant tweet and hence put the condition $p_l \geq q_l$. For example, in Table II the location name *Bondi Beach* appears in all the relevant tweets. Also the *Coherence* property varies in the range [0,1] which allows us to define a *Beta* distribution for the two classes. The motivation behind using the *Beta* distribution is that it is more suitable for a random behavior of proportions. We set the parameters as (α_R, β_R) for $R_j = 1$ and $(\alpha_{\tilde{R}}, \beta_{\tilde{R}})$ for $R_j = 0$. We can now define the conditional probabilities for a tweet T_j using the coherence score and the location names as defined below:

$$P(C_{ij}|R_j = 1) = B(\alpha_R, \beta_R, C_{ij})$$
$$P(C_{ij}|R_j = 0) = B(\alpha_{\tilde{R}}, \beta_{\tilde{R}}, C_{ij})$$
$$P(L|R_j = 1) = \prod_{l=1}^{L} p_l^{L_{lj}} (1 - p_l)^{(1-L_{lj})}$$
$$P(L|R_j = 0) = \prod_{l=1}^{L} q_l^{L_{lj}} (1 - q_l)^{(1-L_{lj})}$$

We use the Expectation-Maximization (EM) algorithm in order to find the relevance (latent variable) of the tweets and also estimate the unknown parameters for the Coherence and location names. Given an observed data X, that is the Instagram tags and location names along with retrieved tweets, one should carefully select the values of the latent variable R and the unknown parameters θ to formulate the likelihood function $f(\theta; X, R) = p(X, R|\theta)$. The EM algorithm finds the maximum likelihood estimate by iteratively performing the following steps:

- E-step: Compute the expected log likelihood function, where the expectation is taken with respect to the computed conditional distribution of the latent variables given the current settings and observed data:

$$Q(\theta|\theta^{(t)}) = E_{R|X,\theta^{(t)}}[logf(\theta; X, R)] \tag{4}$$

- M-step: Find the parameters that maximize the Q function in the E-step to be used as the estimate of θ for the next iteration:

$$\theta^{(t+1)} = argmax\ Q(\theta|\theta^{(t)}) \tag{5}$$

We denote the probability of a tweet being relevant $P(R_j = 1)$ as d. Thus, the set of unknown parameters for the observed data X is given by $\theta = (p_l, q_l, \alpha_R, \beta_R, \alpha_{\tilde{R}}, \beta_{\tilde{R}}, d)$. The likelihood function $f(\theta; X, R)$ is given by:

$$p(X, R|\theta) = \prod_{j=1}^{N} \Bigg\{ \prod_{l=1}^{L} p_l^{L_{lj}} (1 - p_l)^{(1-L_{lj})} \times B(\alpha_R, \beta_R, C_{ij}) \times d \times R_j + \prod_{l=1}^{L} q_l^{L_{lj}} (1 - q_l)^{(1-L_{lj})} \times B(\alpha_{\tilde{R}}, \beta_{\tilde{R}}, C_{ij}) \times (1 - d) \times (1 - R_j) \Bigg\} \tag{6}$$

In eq. (6), d represents the overall prior probability that an arbitrary tweet is relevant. We can now formulate an expectation maximization algorithm that jointly estimates the parameter vector θ and the probability that latent variable $R_j = 1$.

D. Deriving the E-step and M-step

Given the likelihood function as described in eq. (6), we substitute it to the definition of Q function of the Expectation Maximization. Thus the E-step becomes:

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= E_{R|X,\theta^{(t)}}[logf(\theta; X, R)] \\ &= \sum_{j=1}^{N} \Bigg\{ P(R_j = 1|X_j, \theta^{(t)}) \times \Big[\sum_{l=1}^{L} (L_{lj} log p_l + (1 - L_{lj}) log(1 - p_l)) \\ &\qquad + logB(\alpha_R, \beta_R, C_{ij}) + logd \Big] + \\ &\quad P(R_j = 0|X_j, \theta^{(t)}) \times \Big[\sum_{l=1}^{L} (L_{lj} log q_l + (1 - L_{lj}) log(1 - q_l)) \\ &\qquad + logB(\alpha_{\tilde{R}}, \beta_{\tilde{R}}, C_{ij}) + log(1 - d) \Big] \Bigg\} \end{aligned} \tag{7}$$

where X_j is the location names and the hashtag signature A_i associated with a tweet T_j and $P(R_j = 1|X_j, \theta^{(t)})$ is the conditional probability of the latent variable R_j to be true for the given set of observations, which is given by:

$$\begin{aligned} P(R_j = 1|X_j, \theta^{(t)}) &= R(t, j) \\ &= \frac{P(X_j, \theta^{(t)}|R_j = 1)P(R_j = 1)}{P(X_j, \theta^{(t)}|R_j = 1)P(R_j = 1) + P(X_j, \theta^{(t)}|R_j = 0)P(R_j = 0)} \\ &= \frac{U(t, j) \times d}{U(t, j) \times d + V(t, j) \times (1 - d)} \end{aligned} \tag{8}$$

where $U(t, j)$ and $V(t, j)$ are defined as:

$$U(t,j) = \prod_{l=1}^{L} p_l^{L_{lj}} (1-p_l)^{(1-L_{lj})} \times B(\alpha_R, \beta_R, C_{ij}) \tag{9a}$$

$$V(t,j) = \prod_{l=1}^{L} q_l^{L_{lj}} (1-q_l)^{(1-L_{lj})} \times B(\alpha_{\tilde{R}}, \beta_{\tilde{R}}, C_{ij}) \tag{9b}$$

Similarly, $P(R_j = 0|X_j, \theta^{(t)})$ can be represented as:

$$P(R_j = 0|X_j, \theta^{(t)}) = 1 - R(t,j) = \frac{V(t,j) \times (1-d)}{U(t,j) \times d + V(t,j) \times (1-d)} \tag{10}$$

Substituting from eq. (8) and eq. (10) into eq. (7) we get:

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= E_{R|X,\theta^{(t)}}[log f(\theta; X, R)] \\ &= \sum_{j=1}^{N} \Bigg\{ R(t,j) \times \Bigg[\sum_{l=1}^{L} (L_{lj} log p_l + (1 - L_{lj}) log(1-p_l)) \\ &\qquad + log B(\alpha_R, \beta_R, C_{ij}) + log d \Bigg] + \\ &\quad (1 - R(t,j)) \times \Bigg[\sum_{l=1}^{L} (L_{lj} log q_l + (1 - L_{lj}) log(1-q_l)) \\ &\qquad + log B(\alpha_{\tilde{R}}, \beta_{\tilde{R}}, C_{ij}) + log(1-d) \Bigg] \Bigg\} \end{aligned} \tag{11}$$

For the M-step we select θ^* that maximizes $Q(\theta|\theta^{(t)})$. Thus, we set the derivatives $\frac{\partial Q}{\partial p_l} = 0$, $\frac{\partial Q}{\partial q_l} = 0$, $\frac{\partial Q}{\partial \alpha_R} = 0$, $\frac{\partial Q}{\partial \beta_R} = 0$, $\frac{\partial Q}{\partial \alpha_{\tilde{R}}} = 0$, $\frac{\partial Q}{\partial \beta_{\tilde{R}}} = 0$, and $\frac{\partial Q}{\partial d} = 0$. With respect to d we have the following equation:

$$\sum_{j=1}^{N} \frac{R(t,j)}{d} + \sum_{j=1}^{N} \frac{(1 - R(t,j))}{1-d} \tag{12}$$

Solving the eq. (12) we get the following value of d:

$$d^{(t+1)} = d^* = \frac{\sum_{j=1}^{N} R(t,j)}{N} \tag{13}$$

Since we have an inequality defined with respect to p_l and q_l, we use the Karush-Kuhn-Tucker (KKT) conditions while performing the maximization step. Thus our inequality constraint $(g : q_l - p_l \leq 0)$ allows us to define two regions depending on whether the constraint is inactive or active. In the case where g is inactive the Lagrangian multiplier (λ) will have a value 0 and we get the following equations:

$$\sum_{j=1}^{N} \left[R(t,j) \left(\frac{L_{lj}}{p_l^*} - \frac{1 - L_{lj}}{1 - p_l^*} \right) \right] = 0 \tag{14a}$$

$$\sum_{j=1}^{N} \left[(1 - R(t,j)) \left(\frac{L_{lj}}{q_l^*} - \frac{1 - L_{lj}}{1 - q_l^*} \right) \right] = 0 \tag{14b}$$

Solving the above set of equations we get the following values of p_l^* and q_l^*:

$$p_l^{(t+1)} = p_l^* = \frac{\sum_{L_{lj}=1} R(t,j)}{\sum R(t,j)} \tag{15a}$$

$$q_l^{(t+1)} = q_l^* = \frac{K_l - \sum_{L_{lj}=1} R(t,j)}{N - \sum R(t,j)} \tag{15b}$$

where K_l is the total number of tweets in which location name L_l is present. However, if the constraint is not satisfied and we are in the active region, then we need to solve for the Lagrangian multiplier subject to the condition $\lambda \geq 0$. By solving for the optimal values, we get the following equation:

$$p_l^* = q_l^* = \frac{\sum_{j=1}^{N} R(t,j)}{N} \tag{16}$$

From the above equation we see that p_l and q_l have the same value, which indicates that the location name does not have a pivotal role in determining the relevancy of the tweet. For the *Beta* distribution parameters, we get the following set of equations:

$$\psi(\alpha_R^*) - \psi(\alpha_R^* + \beta_R^*) = \frac{1}{N} \sum_{j=1}^{N} R(t,j) log C_{ij} \tag{17a}$$

$$\psi(\beta_R^*) - \psi(\alpha_R^* + \beta_R^*) = \frac{1}{N} \sum_{j=1}^{N} R(t,j) log(1 - C_{ij}) \tag{17b}$$

$$\psi(\alpha_{\tilde{R}}^*) - \psi(\alpha_{\tilde{R}}^* + \beta_{\tilde{R}}^*) = \frac{1}{N} \sum_{j=1}^{N} (1 - R(t,j)) log C_{ij} \tag{17c}$$

$$\psi(\beta_{\tilde{R}}^*) - \psi(\alpha_{\tilde{R}}^* + \beta_{\tilde{R}}^*) = \frac{1}{N} \sum_{j=1}^{N} (1 - R(t,j)) log(1 - C_{ij}) \tag{17d}$$

In order to find the optimal values of the parameters, we use the Newton-Raphson method on the above set of equations. The work described in [16] covers the Newton-Raphson method derivation for maximum likelihood estimation. Once we have relevance computed as a probability value, we can next run the event detection technique for Twitter as well. For every Instagram cluster, we say that the event is true if it has a corresponding set of relevant tweets and for every Twitter cluster we only retain the tweets that got classified as relevant. In this way, we achieve our goal of corroborating the events detected on both the networks.

E. Final Algorithm

Given an Instagram cluster containing a set of hashtags and location information we first retrieve the tweets based on the tag similarity metric. We then initialize the value of the parameters to some random values. For our experiments we assign $d = 0.3$, $\alpha_R = 2$, $\beta_R = 1$, $\alpha_{\tilde{R}} = 1$, $\beta_{\tilde{R}} = 2$, $p_l = 0.6$,

and $q_l = 0.3$. The algorithm then performs the E-steps and M-steps iteratively until θ converges. Specifically, at every E-step we try to determine the probability value of a tweet T_j being relevant as assign it to $R(t, j)$. Based on this probability value we next perform M-step where we identify the optimal value of all the parameters as described in our derivation. After convergence, we get a ranked list of tweets based on the $R(t, j)$ values. Alternatively we can also assign a binary value to the tweets based on the condition $R = 1$ if $R(t, j) \geq 0.5$ or $R = 0$ otherwise. The pseudo code is shown in algorithm 1

Algorithm 1 Fusion Model

procedure EM ALGORITHM
Initialize θ with random values as mentioned in the above section
  while $\theta^{(t)}$ does not converge **do**
    for $j = 1 : N$ **do**
      Compute R(t,j) based on Equation 8
    end for
    $\theta^{(t+1)} = \theta^{(t)}$
    for $l = 1 : L$ **do**
      Compute $p_l^{(t+1)}$, $q_l^{(t+1)}$ *based on Equations 15, 16*
      Update $p_l^{(t)}$, $q_l^{(t)}$ *with* $p_l^{(t+1)}$, $q_l^{(t+1)}$ *in* $\theta^{(t+1)}$
    end for
    Compute $d^{(t+1)}$ *based on Equation 13*
    Update $d^{(t)}$ *with* $d^{(t+1)}$ *in* $\theta^{(t+1)}$
    Compute Beta distribution parameters using Equation 17
    Update Beta distribution parameters in $\theta^{(t+1)}$
    $t = t + 1$
  end while
  Let $R(t, j)^* =$ *converged value of* $R(t, j)$
  for $j = 1 : N$ **do**
    if $R(t, j)^* \geq 0.5$ **then**
      $R_j = 1$
    else
      $R_j = 0$
  end for

F. Algorithm Analysis

In this section we provide a brief overview of the complexity of algorithm 1. Firstly, the threshold setting for $R(t, j)^*$ is approximated on the basis of Coherence score distribution as shown in Figure 6(b) where a high fraction of true negatives are below the value of 0.5 score. In eq. (11) we consider the maximum likelihood estimation of parameters for some independent binomial distributions and two independent beta distributions. One can simply confirm that the Hessian is negative definite which proves that the function is concave. As for the EM algorithm it is well known that the convergence is guranteed [10] but it may converge to a local minima. The time complexity for the binomial parameters is $O(N)$ while the beta parameters is dependent on the required precision during the Newton Raphson iterations. The overal complexity depends on the number of EM iterations for convergence.

IV. EVALUATION

Our evaluation is divided into two sections. The first one is a simulation study on a synthetic data which allows us to verify the formulated Expectation-Maximization (EM) approach. The second one is an actual experiment on the *Social Fusion* service using real world dataset. The details of both the experiments are presented below.

A. Simulation Study

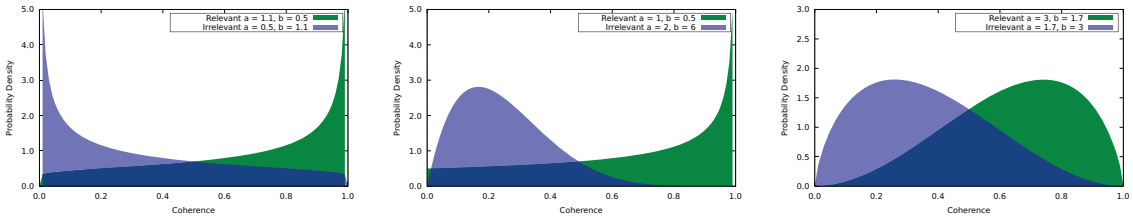
We use *Python* programming language to code the simulator and the final algorithm. The number (N) of tweets is varied between $\{100, 200, 500, 1000\}$ and the *Coherence* score is obtained using the Beta distribution. For every N tweets, we also pre-define the fraction (d) of tweets that will be labeled as relevant ($R = 1$). The value of d is varied between $\{0.1, 0.2, 0.5\}$. We also use three location names for every run of the simulation. The values for the parameters associated with location names are shown in Table IV. For the case of *Street Name* we expect the tweet to be more likely relevant while *State* has an equal chance of being present in relevant and irrelevant tweets. For every N selected tweets we select the fraction d that will be marked as relevant. Once we have the ground truth available for the tweets we next use the the location specifier parameters to link the location names with the tweets depending on the relevance value. Finally we assign the *Coherence* score using the Beta distribution as described above.

TABLE IV: Location specifier for simulation

Type (L_l)	p_l	q_l
Street Name	0.8	0.2
City	0.6	0.3
State	0.5	0.5

Figure 2 shows the different parameter (α, β) settings that we consider to generate the *Coherence* score of a tweet given the label. The x-axis is the *Coherence* score and the y-axis is the probability density. The region in *green* color represents the relevant tweet scores and the region in *blue* color represents the irrelevant tweet scores. Parameter setting I is the case where majority of the relevant tweets are concentrated towards the high coherence score and majority of the irrelevant tweets are concentrated towards the low coherence score. Parameter setting II is the case where we keep relevant tweet score distribution same as setting I but change the irrelevant tweet score distribution slightly towards a moderate score range. Finally Paramete setting III is the case where there is a significant overlap between the two distributions. In a real world environment, it would not be surprising to observe this kind of distribution. For each parameter setting, we run our algorithm and compare the expected labels with the original labels. We use three metrics - Precision, Recall, and Accuracy to measure the performance. For every combination of N, d, and *Coherence* parameter settings we run the simulator and the algorithm 10 times and take the average value for each metric.

Figure 3 is the metric evaluation plot for simulation using the parameter setting I. The first subplot is for precision, which measures the fraction of expected relevant tweets that are correctly labeled (as relevant). The x-axis represent the number of tweets with d fraction of tweets labeled as relevant. The y-axis represents the average precision value over 10 runs for



(a) Parameter Setting I (b) Parameter Setting II (c) Parameter Setting III

Fig. 2: Different Parameter Settings on Beta Distribution for Coherence

the corresponding settings. The second subplot is for recall, which measures what fraction of relevant tweets that have been identified as such. The x-axis represents the number of tweets with d fraction of tweets labeled as relevant and the y-axis represents the average recall value over 10 runs for the corresponding setting. The third subplot shows the accuracy of the overall algorithm at correctly labeling the relevant and irrelevant tweets. Figure 4 is the metric evaluation plot for simulation using the parameter setting II and figure 5 is the metric evaluation plot for simulation using the parameter setting III. For the first two parameter settings, precision and accuracy of the model are well above 90% and recall is above 80% on average. The third parameter setting, which has a significant overlap in the *Coherence* distribution between the two classes generates slighlty lower values in terms of precision and recall compared to the previous parameter settings.

TABLE V: Average error in parameter estimation

Parameter	Average Error
d	0.0122
p_l, q_l	0.0103, 0.0292
α_R, β_R	0.0231, 0.0366
$\alpha_{\tilde{R}}$, $\beta_{\tilde{R}}$	0.0128, 0.0488

In addition to the above comparisons, we also determine the average error in the estimation of the fraction of tweets d, location name parameters, and the Beta distribution parameters used for *Coherence* score. Table V indicates the average error values over all the runs with different combinations of parameter and tweet count N settings. The average error in estimating the value for different parameters is well within 0.05. Thus, with the help of simulation experiments, we have established the fact that our fusion model using the EM algorithm is very good at identifying the relevance of a given set of tweets associated with an Instagram location and hashtags. It remains to verify that this is indeed the case with real Twitter and Instagram data, which is the topic of the next section.

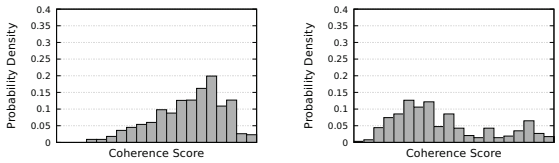
B. Dataset Experiments

In this section, we discuss evaluation using a real world dataset. We conduct this experiment on the *Social Fusion* service that we implemented to run the fusion algorithm. The *Task Info* file was provided with the keyword "*protest*" which initiated the crawler module to collect data from both Twitter and Instagram. We specifically consider the data logged on to the disk for the time duration February 1, 2016 to February 29, 2016. Table VI summarizes the data collected during this period. For each row we show the total number of tweets, the fraction of tweets that are geotagged (tweets with latitude and longitude information available), and the number of Instagram posts. We retain only those Instagram posts that have location information available.

TABLE VI: Statistics of collected datasets

Dataset	# Tweets	Geotag Fraction	Instagram posts
Feb 2016 Week 1	77001	0.0016	1377
Feb 2016 Week 2	78334	0.0012	1424
Feb 2016 Week 3	75639	0.0015	1489
Feb 2016 Week 4	64669	0.0015	1398

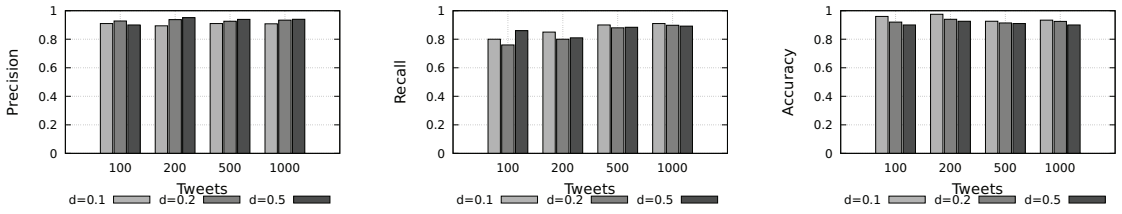
We first show that our *Coherence* metric indeed meaningfully distinguishes relevant and non-relevant tweets (to a given Instagram post). To do so we consider an arbitrary sample of clusters generated by the Instagram Event Localizer module along with the candidate tweets. For each cluster we manually label the tweets as relevant and non-relevant. We then generate a frequency distribution of the respective *Coherence* scores which is shown in figure 6, where the left subplot corresponds to the relevant tweet scores and the right subplot corresponds to the non-relevant tweet scores. It can be observed that we have two different Beta-like distributions that can be approximated using our model. This plot validates our model assumptions regarding the distribution of *Coherence* of relevant and non-relevant tweets.



(a) Relevant Tweets (b) Non-relevant Tweets

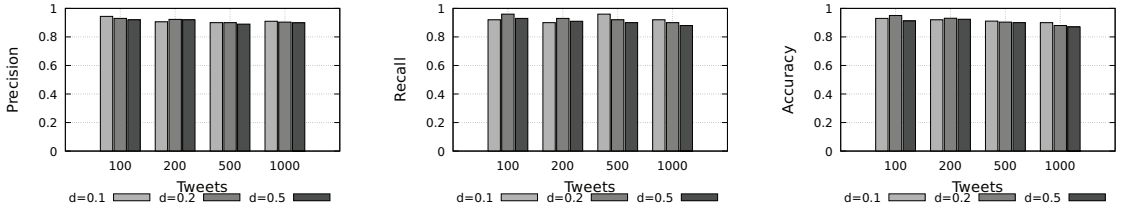
Fig. 6: Frequency distribution for Coherence scores

In order to evaluate the performance of our fusion model at event detection, we select two individual event detection techniques for each Instagram and Twitter. The Points of Interest (POI) method described in [20] and the Instagram Event localization (InstaLoc) [13] are used for detecting events on Instagram dataset. The Earthquake detection (TweetEvent) [19]



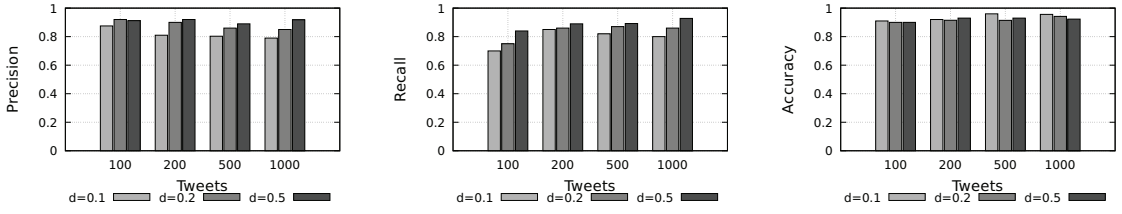
(a) Precision (b) Recall (c) Accuracy

Fig. 3: Evaluation plots for simulation using Parameter Setting I



(a) Precision (b) Recall (c) Accuracy

Fig. 4: Evaluation plots for simulation using Parameter Setting II



(a) Precision (b) Recall (c) Accuracy

Fig. 5: Evaluation plots for simulation using Parameter Setting III

and ClariSense [11] are used for detecting events on Twitter dataset. The evaluation is done using two separate criteria. The first one is the improvement in the amount of detected events against the Instagram detection techniques itself. The second one is the fraction of false positives present in the data against Twitter based detection techniques.

For both the Instagram event detection techniques, we eliminate the below threshold clusters as mentioned in the respective papers but do not follow the same while applying the social fusion method. This allows us to see if the clusters that got eliminated due to lack of support can actually be identified using the fusion method. At the same time we use both the Twitter detection techniques with the same parameters mentioned in the papers, and for the fusion method we only retain those clusters that contain any relevant tweets. With the mentioned techniques we have four pairs of baselines - (i) POI and TweetEvent (B1), (ii) InstaLoc and TweetEvent (B2), (iii) POI and ClariSense (B3), and (iv) InstaLoc and ClariSense (B4).

Figure 7 shows the plot for comparison with and without the fusion model for each of the baseline methods in order to find the improvement in the number of total events considering only Instagram detection techniques. There are four subplots for each week in the dataset with x-axis representing the baseline method and the y-axis representing the total detected events. It is evident from the plot that with the help of fusion model we are able to detect more events in general for any selected baseline method. Figure 8 shows the plot for comparison with and without the fusion model for each of the baseline methods in order to find the precision considering only Twitter detection techniques. There are four subplots for each week in the dataset with x-axis representing the baseline method and the y-axis representing the precision. This plot shows that with the help of fusion model we are able to remove a significant amount of false positives thereby resulting in a higher precision.

The results substantiate the contribution claim made in this paper. Namely, the new fusion based technique offers a better trade-off between false-positives and false-negatives attained using techniques that exploit individual networks separately. We offer significantly fewer false-positives than Twitter-based detection, and significantly fewer false-negatives (i.e., more true positives) than Instagram-based detection, thereby attain-

ing a new point in the aformentioned trade-off space.

V. Related Work

There has been a huge surge in using social networks for sharing content in real time related to physical event observations. This activity is similar to sensing where users provide their sensory readings in the form of text, pictures, and video. Past works [19], [9], [7] have demonstrated that such events can indeed be detected using techniques that try to model the behavior of the pattern of extracted features before, during, and after the events. One such technique, is described by the authors of [25] where they apply wavelet analysis on the raw frequency of the words used on Twitter stream and then remove trivial words using the signal correlation. In our previous work [11], we showed how Twitter posts related to traffic incidents can be correlated to the anomalies observed in the physical sensors on the road networks. A few papers have also focused on using the geo-tag information available within the content in order to find clusters that have unusual behavior compared to a stored history within a spatial region. A recent approach [23] monitors Twitter posts within a geographic region and then uses a supervised approach to classify true events. However, the amount of geo-tagged data available is far less than the actual volume of data. [28] is another new work that tries to use the geo tagged tweets to detect emerging topical words in a spatial domain and design a language model for the future. In addition to the above efforts for detecting the events much work has been done towards finding the credibility of the information propagated in social networks. [8] uses features extracted from tweets in order to classify them as true or a rumor. The work mentioned in [14] also focuses to find the credibility of the events in Twitter network using a PageRank-like credibility propagation. These and many other works towards finding the truthness of the information provide a good way to reduce the spams and false positives within the network but it still requires a lot of modeling using the already existing data.

Instagram is another social network where people post pictures and videos with a higher percentage of geo-tag compared to Twitter. The use of locations by the users tends to deliver much credible information. However the amount of such data available is less but considerably higher than Twitter. Various event detection techniques using Instagram have also been studied in the past. One such work described by the authors of [27] has been promising for monitoring city level local events. [20] is the earliest work that uses Instagram to study the urban social behavior and the city dynamics. In our own recent work [13] we showed how to identify events for urban spaces in an unsupervised way.

Contrary to all the previous approaches not much work has been done in fusing the same entities (or events) detected in multiple networks thereby enhancing the overall credibility of the events. The work described by authors of [26] considers Twitter and Instagram data to detect and summarize events but they rely on supervised techniques along with geo-location information. However in this paper we aim at improving event detection by fusing data across multiple networks without any dependence on historical data and detect events with varying degree of popularity. Such events can be effectively retrieved by our fusion method provided enough correlation exists between the data posts on different networks, even when it is hard to detect them by analyzing each network independently. Our work provides an important means to fill the gap in identifying and corroborating the events present in multiple networks.

VI. Conclusions

This paper describes a service which uses a fusion model for integrating data from two different social media platforms, namely, *Twitter* and *Instagram*. The work offers a better trade-off between false-positives and false-negatives compared to approaches that utilize individual networks independently. Specifically, we show that we offer fewer false positives compared to Twitter and fewer false negatives compared to Instagram, offering a new point on the trade-off curve. The motivation for our work comes from the fact that many events offer signatures in multiple networks that can somehow be correlated with the help of intrinsic characteristics such as location mentions and coherence among event descriptions. We design an algorithm that is capable of fusing content from Twitter and Instagram in an unsupervised way. We first study the validity of our model using simulations and evaluate the performanace using precision and recall metrics. Finally, we use real world datasets to confirm the advantages of the fusion approach. We would like to mention that even though our experiments are based on two specific social networks but the same approach can be generalized to other social networks provided they share some feature space (such as words) to generate potential candidates from one of the networks. The challenge to deal with the credibility of user generated data can be solved using a past work [24] to find the truthness of the claims.

VII. Acknowledgements

Research reported in this paper was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement W911NF-09-2-0053, DTRA grant HDTRA1-10-1- 0120, and NSF grants NSF CNS 13-29886, CNS 09-58314, and CNS 10-35736. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] Google maps api. https://developers.google.com/maps/documentation/geocoding/.

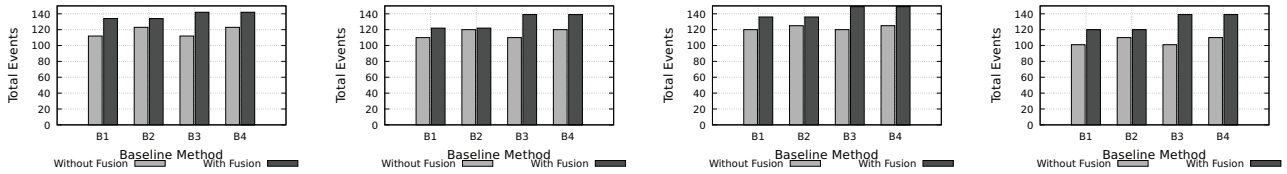
[2] Instagram. https://www.instagram.com.

[3] Instagram geotag statistic. http://neilpatel.com/2016/09/26/4-easy-ways-to-generate-ecommerce-sales-from-instagram/.

[4] Twitter. http://www.twitter.com/.

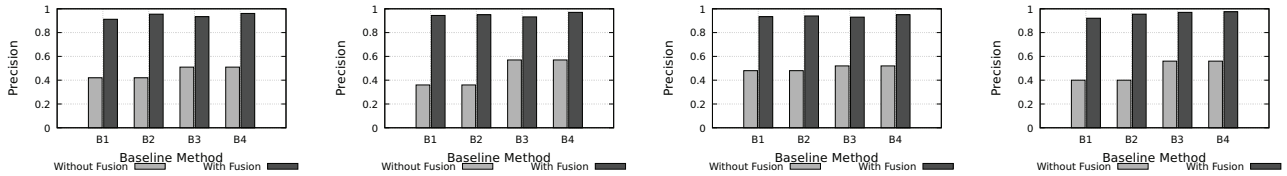
[5] Twitter online statistics. http://www.internetlivestats.com/twitter-statistics/.

[6] M. T. A. Amin, S. Li, M. R. Rahman, P. T. Seetharamu, S. Wang, T. Abdelzaher, I. Gupta, M. Srivatsa, R. Ganti, R. Ahmed, and H. Le. SocialTrove: A self-summarizing storage service for social sensing. In *International Conference on Autonomic Computing (ICAC'15)*, pages 41–50. IEEE, July 2015.



(a) Week 1 (b) Week 2 (c) Week 3 (d) Week 4

Fig. 7: Instagram detection improvement - comparison of different baseline methods with and without fusion method



(a) Week 1 (b) Week 2 (c) Week 3 (d) Week 4

Fig. 8: Twitter detection precision - comparison of different baseline methods with and without fusion method

[7] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 291–300, New York, NY, USA, 2010. ACM.

[8] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 675–684, New York, NY, USA, 2011. ACM.

[9] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 143–152. IEEE, 2012.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[11] P. Giridhar, M. T. Amin, T. Abdelzaher, L. Kaplan, J. George, and R. Ganti. Clarisense: Clarifying sensor anomalies using social network feeds. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pages 395–400. IEEE, 2014.

[12] P. Giridhar, S. Wang, T. Abdelzaher, R. Ganti, L. Kaplan, and J. George. On localizing urban events with instagram. In *IEEE Infocom*.

[13] P. Giridhar, S. Wang, T. Abdelzaher, R. Ganti, L. Kaplan, and J. George. On localizing urban events with instagram. In *IEEE Infocom, Atlanta, GA, May 2017*, 2017.

[14] M. Gupta, P. Zhao, and J. Han. *Evaluating Event Credibility on Twitter*, pages 153–164.

[15] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang. Tedas: A twitter-based event detection and analysis system. In *2012 IEEE 28th International Conference on Data Engineering*, pages 1273–1276. IEEE, 2012.

[16] C. E. B. Owen. Parameter estimation for the beta distribution. 2008.

[17] P. Paraskevopoulos and T. Palpanas. Where has this tweet come from? fast and fine-grained geolocalization of non-geotagged tweets. *Social Network Analysis and Mining*, 6(1):89, 2016.

[18] G. Petkos, S. Papadopoulos, V. Mezaris, R. Troncy, P. Cimiano, T. Reuter, and Y. Kompatsiaris. Social event detection at mediaeval: a three-year retrospect of tasks and results. In *ICMR 2014 Workshop on Social Events in Web Multimedia (SEWM)*, pages 27–34, 2014.

[19] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[20] T. H. Silva, P. O. V. de Melo, J. M. Almeida, J. Salles, and A. A. Loureiro. A picture of instagram is worth more than a thousand words: Workload characterization and application. In *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 123–132. IEEE, 2013.

[21] T. H. Silva, P. O. Vaz de Melo, J. M. Almeida, J. Salles, and A. A. Loureiro. A comparison of foursquare and instagram to the study of city dynamics and urban social behavior. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, page 4. ACM, 2013.

[22] M. Thakur, D. Patel, S. Kumar, and J. Barua. Newsinstaminer: Enriching news article using instagram. In *International Conference on Big Data Analytics*, pages 174–188. Springer, 2014.

[23] M. Walther and M. Kaisser. *Geo-spatial Event Detection in the Twitter Stream*, pages 356–367. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[24] D. Wang, M. T. Amin, S. Li, T. Abdelzaher, L. Kaplan, S. Gu, C. Pan, H. Liu, C. C. Aggarwal, R. Ganti, X. Wang, P. Mohapatra, B. Szymanski, and H. Le. Using humans as sensors: An estimation-theoretic perspective. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pages 35–46, April 2014.

[25] J. Weng and B.-S. Lee. Event detection in twitter. *ICWSM*, 11:401–408, 2011.

[26] C. Xia, J. Hu, Y. Zhu, and M. Naaman. What is new in our city? a framework for event extraction using social media posts. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 16–32. Springer, 2015.

[27] C. Xia, R. Schwartz, K. Xie, A. Krebs, A. Langdon, J. Ting, and M. Naaman. Citybeat: Real-time social media visualization of hyper-local city data. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 167–170, New York, NY, USA, 2014. ACM.

[28] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han. Geoburst: Real-time local event detection in geo-tagged tweet streams. SIGIR 2016, May 2016.