Unsupervised Event Tracking by Integrating Twitter and Instagram

Shiguang Wang, Prasanna Giridhar
University of Illinois
Urbana, IL 61801
(swang83,giridha2)@illinois.edu

Lance Kaplan
US Army Research Laboratory
Adelphi, MD 20783
lance.m.kaplan@us.army.mil

Tarek Abdelzaher
University of Illinois
Urbana, IL 61801
zaher@illinois.edu

ABSTRACT

This paper proposes an unsupervised framework for tracking real world events from their traces on Twitter and Instagram. Empirical data suggests that event detection from Instagram streams errs on the false-negative side due to the relative sparsity of Instagram data (compared to Twitter data), whereas event detection from Twitter can suffer from false-positives, at least if not paired with careful analysis of tweet content. To tackle both problems simultaneously, we design a unified unsupervised algorithm that fuses events detected originally on Instagram (called I-events) and events detected originally on Twitter (called T-events), that occur in adjacent periods, in an attempt to combine the benefits of both sources while eliminating their individual disadvantages. We evaluate the proposed framework with real data crawled from Twitter and Instagram. The results indicate that our algorithm significantly improves tracking accuracy compared to baselines.

ACM Reference format:
Shiguang Wang, Prasanna Giridhar, Lance Kaplan, and Tarek Abdelzaher. 2017. Unsupervised Event Tracking by Integrating Twitter and Instagram. In *Proceedings of The 2nd International Workshop on Social Sensing, Pittsburgh, PA USA, April 2017 (SocialSens'17),* 6 pages.
DOI: http://dx.doi.org/10.1145/3055601.3055615

1 INTRODUCTION

In this paper, we investigate the problem of tracking events in physical spaces with the help of data shared on social networks by users observing them. As per the recent statistics [1, 2], Twitter has 317 million monthly active users and more than 500 million uploaded tweets per day. Instagram has 600 million monthly active users and more than 80 million uploaded images per day. With the constant increase in users and content, these social networks are becoming a great source of crowdsourced information.

This work contributes to literature on event detection from social media. Specifically, we investigate the degree to which event detection can be improved by fusing data from Twitter and Instagram. The fusion operation itself is separable from the individual event detection techniques used in each network. While, in this workshop publication, we demonstrate a proof of concept by fusing outputs of two specific detection techniques, in principle, we aim at a fusion algorithm that is independent of per-network detection specifics.

One robust observation across several Twitter-based detection techniques, compared by the authors [17], is that it is difficult to filter out false positives. This might be attributed to the vast number of posted tweets, which increases the likelihood of formation of spurious clusters, not representative of actual physical events. Instagram, on the other hand, has sparser content, which leads to a much higher precision (when clustered for event detection), although a lower recall [6]. That is to say, clusters of pictures detected from Instagram posts are generally indicative of real geo-events, although (due to the smaller number of posts) more events are missed. To improve Instagram recall, recent work proposed techniques for corroborating Instagram pictures using Twitter data to distinguish one-off irrelevant pictures from those related to actively discussed events.[1] The approach allows detection of events with smaller support in Instagram, but does not detect events that have *no* Instagram presence.

In contrast, this paper aims at fusing events from Twitter and Instagram in a manner that chooses the best of both sets. Importantly, it includes events with representation on only one of the two networks (i.e., those on Twitter only or Instagram only), as long as they are sufficiently supported. We show that the approach offers a better trade-off between precision and recall. Note that, events discussed on social media are not a flat structure. Rather, they can be split into sub-events or aggregated into larger events. For example, social media users might discuss specific demonstrations at different locations. They might also refer to the larger context that brings about these protests, and to different individual incidents that occurred during one demonstration. By combining clusters from both Instagram and Twitter one has a higher chance of uncovering such event linkages, hence consolidating valid threads (tracks) of events over time, and eliminating poorly supported events outside such threads.

The rest of this paper or organized as follows. In Section 2 we present the problem formulation and the design of our approach. The evaluation is discussed in Section 3. Related work is described in Section 4. Finally, conclusions are presented in Section 5.

2 PROBLEM DEFINITION AND SYSTEM DESIGN

In this section, we first define our problem of event tracking integrating Instagram and Twitter data, then we overview our system

SocialSens'17, Pittsburgh, PA USA
© 2017 ACM. 978-1-4503-4977-2/17/04…$15.00
DOI: http://dx.doi.org/10.1145/3055601.3055615

[1]This work is currently under submission and can be accessed via the link http://hdl.handle.net/2142/95127 [5].

architecture and design, and introduce each of the system modules in detail.

2.1 Problem Formulation

We study the problem of integrating Twitter and Instagram to detect physical events and to track them as they evolve. Our goal is to do so online, using a language-agnostic, unsupervised approach.

Following previous work [17], time is discretized into slots called *windows*. Each event instance, E_i, has a detection (or start) time, and a termination (or finish) time. The event is said to be ongoing between these two times. Each event instance is further associated with a chronologically sorted list of timestamped tweets and Instagraph photos that describe it up to the current time.

We restrict our attention to approaches that do not analyze content of tweets or images. This is because we want our detection and tracking system to be generally applicable regardless of language. Also, due to the heavy cost of human labelling of text and pictures, an unsupervised approach is desired.

We define our problem as follows: given the stream of tweets on Twitter and photo posts on Instagram in window k, for each k

- determine the set of ongoing event instances at time k, and
- identify the set of tweets and images related to each event instance.

2.2 System Architecture and Design

The system architecture is illustrated in Figure 1, where the boxes denote system components and arrows denote data flows. The input of the system comprises the data crawled from Twitter and Instagram, and the output comprises tweets and images associated with every event instance. We continuously crawl Twitter text and Instagram photo posts and feed the data to our system, say once per hour. In the system, we have two event detectors, the T-detector and I-detector, respectively. The T-detector detects events from Twitter data using a recently proposed detection algorithm [17]. The I-detector detects events from Instagram. We use an enhanced version [5] that corroborates detected potential Instagram events using tweets with similar tags. The events detected by the T-detector are called *T-events (or T-buckets)*, and those detected by the I-detector are called *I-events (or I-buckets)*. Both the T-events and I-events are fed into the fusion module, where we correlate their buckets. The fused events are then fed into the tracking module, where we track these events over time in an online fashion by consolidating buckets of same events over time. Finally, we display a summary of current and past events on demand. Next, we describe each of the system modules in detail.

2.3 Event Detection

The T-detector, in our system, was first introduced in the Storyline paper [17]. The high-level idea is that, in each window, we detect patterns of keywords that occur with disproportionately high frequency compared to the previous window, which translates into high information gain. Evaluation of Storyline [17] demonstrates that this detector does better than other Twitter-based techniques at event demultiplexing. Furthermore, the Storyline detector can be applied online to streaming data, is unsupervisded, and does not require analysis of text.

Since we aim to design a fusion technique that is independent of the specifics of individual detectors, we introduce an independent filtering module after the T-detector to remove false positives. The idea behind the filtering module is that real events tend to focus discussion around a narrower scope of topics, leading to a skew in the distribution of keywords in tweets related to the discussed event. A smaller number of keywords get more frequently mentioned, followed by a tail of infrequent keywords. In contrast, when discussion is not focused, the distribution of keywords is much more evenly spread.

To exploit the above intuition, for each token in the tweet cluster returned by the Twitter-based detector (the T-bucket), we calculate the fractions (empirical probability) of tweets in the cluster containing the token. We then fit the empirical probability distribution of tokens to a Beta distribution, and compute its parameters, α and β. These parameters are then used to detect false positives based on an appropriate threshold.

In order to determine the best threshold, we manually labeled 700 detected events with true event or false positive. The mean values of the Beta parameters for true events were $\alpha = 0.2104$ and $\beta = 0.3987$, and those for false positives were $\alpha = 0.4025$ and $\beta = 394.6$. These values clearly indicate that for true events, there is a subset of tokens that appear in a large potion of tweets in the event cluster, but for false positives, the tokens are distributed much more "evenly" across tweets in each event cluster. To find a single threshold, we further plot the CDF of the *expectation* of the fitted Beta distributions for both the true events and false positives, shown in Figure 2. From the figure, we observe that the expected value of the Beta distribution is a good feature to separate the true events from false positives, as they differ significantly in this feature. Accordingly, we set the threshold to 0.03; if the Beta expectation is greater than this threshold, the bucket is classified as a true event. Otherwise, it is classified as a false positive. Note that, this classifier can work with the output of any tweet clustering scheme.

Our I-detector uses Instagram to detect events, as described in recent work [5]. We use a detector that (1) clusters Instagram photo posts that are co-located in the same proximity and are within a short period of time, and (2) tries to find supporting tweets on Twitter by correlating their hashtags and locations (converting Instagram geotags into street addresses and looking for tweets with corresponding keywords). If we are able to find supporting tweets for a cluster, then we claim successful detection of one (Twitter-corroborated) Instagram event. Please note that the supporting tweets here are not necessarily from some T-bucket(s) that generated by the T-detector. In essence, this approach embeds its own false-positive elimination by seeking corroboration with Twitter. It can be used on top of any other I-detector that returns images related to potential detected events. Note that, the approach returns only those events that have representation in Instagram data. Hence, it returns considerably fewer events compared with the T-detector, although the precision is higher.

Next, we present a fusion algorithm that combines I-events and T-events in a manner that attains both high precision and high recall.

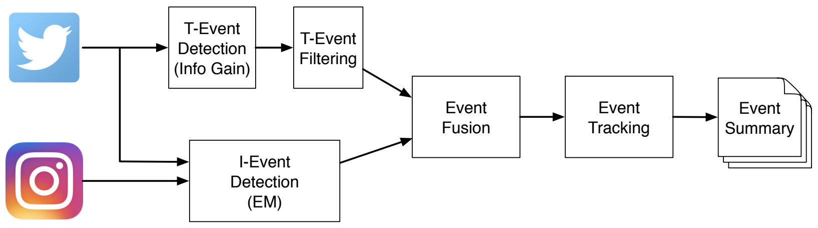

Figure 1: Event Tracking System Architecture

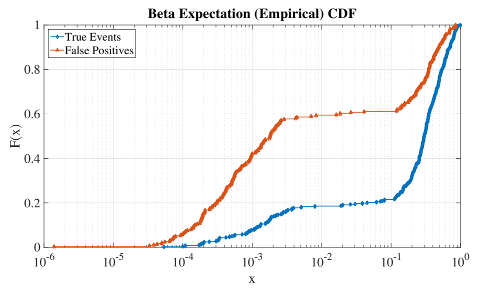

Figure 2: CDF of empirical Beta expectation comparison between true events and false positives

2.4 Event Fusion

The fusion algorithm uses the I-buckets and T-buckets as input. Note that, we have both the tweets and photo metadata in the I-buckets. As mentioned above, if a different I-detector is used that returns only buckets of photo metadata, we can always use the approach in [5] to corroborate these buckets with tweets, thereby augmenting them with related tweets. The effect of such corroboration is to reduce false positives (such as "selfies" and other images not corresponding to events of interest).

Next, we create a bipartite graph $BG = \{I, T, E\}$, where each node $i \in I$ is corresponding to an I-bucket, and each node $t \in T$ is corresponding to a T-bucket. When the similarity in tweets between an I-bucket, i, and a T-bucket, t, is beyond a similarity threshold τ_{bucket}, we add an edge (i, t) to E. The similarity of two buckets is defined by the Jaccard distance between the tweet text tokens of them[2]. Here, the threshold τ_{bucket} denotes whether we want to merge the I-bucket and T-bucket (if similarity above it) or not (otherwise). Hence, for any edge $e_{i,t} \in E$, we merge the corresponding I-bucket i and T-bucket t. We summarize the fusion procedure in Algorithm 1. The time complexity of the algorithm is determined by the size of the bipartite graph, therefore, it is $O(MN + M + N) = O(MN)$ where $M = |I|$ and $N = |T|$.

2.5 Event Tracking and Summary

The event fusion algorithm is then extended to event tracking in a straightforward manner; instead of correlating I-buckets and T-buckets in the same window k as done in the event fusion module, in tracking, we correlate buckets detected in window k and those detected in the previous window $k - 1$, merging those with a high Jaccard similarity. The unified buckets fusion algorithm for both the event fusion module and tracking module simplifies the system implementation and improves the system maintainability. Inspired by Storyline [17], we use a sliding window, such that windows k and $k-1$ overlap as illustrated in Figure 3. In a typical configuration, a window of size 6 hours slides 1 hour at a time.

The output of merged buckets after fusion and tracking generates larger clusters. Each such cluster is associated with a unique ID. Buckets associated with a given ID can be displayed on demand, forming a chronological list of tweets and images that describe event evolution.

[2]Note that I-buckets have associated tweets.

Algorithm 1 Fusion I-bucket and T-bucket

Input: I-buckets and T-buckets in window k, threshold τ_{bucket}
Output: The fusioned buckets.

```
1:  Build the bipartite graph $BG$ with empty edge set $E$
2:  for Each pair of I-bucket $i$ and T-bucket $t$ do
3:      if The similarity scrore is beyond $\tau_{bucket}$ then
4:          $E \leftarrow E + (i, t)$
5:      end if
6:  end for
7:  for Each node $i \in I$ do
8:      Merge each node $t \in T$ s.t. $(i, t) \in E$, and denote the
    merged node as $m_t$
9:      Update $E$ such that $\forall(j, t) \in E, \forall t \in \{m_t\}, E \leftarrow E + (j, m_t)$
10:     Remove every original T-nodes $t$ in any merged node $m_t$
    and remove all edges incident with it, i.e.
    $\forall t \in \{m_t\}, T \leftarrow T - t, E \leftarrow E \setminus \{(., t)\}$
11: end for
12: for Each node $t \in T$ do
13:     Merge each node $i \in I$ s.t. $(i, t) \in E$, and denote the
    merged node as $m_i$
14:     Remove every original I-nodes $i$ in any merged node $m_i$
    and remove all edges incident with it, i.e.
    $\forall i \in \{m_i\}, I \leftarrow I - i, E \leftarrow E \setminus \{(i, .)\}$
15: end for  ▷ (Now bipartite graph $BG$ becomes a match, that is
    $\forall i_0 \in I$ at most one $t \in T$ s.t. $(i_0, t) \in E$ and $\forall t_0 \in T$ at most
    one $i \in I$ s.t. $(i, t_0) \in E$)
16: Merge the matched pairs, and output the merged buckets and
    individual I- or T-buckets
```

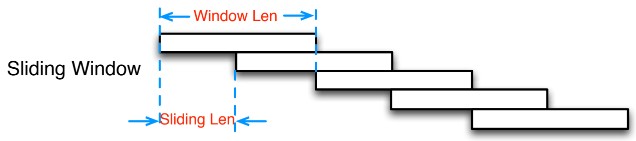

Figure 3: Sliding window

3 EVALUATION

In this section, we first introduce the real-world dataset used in the evaluation and then discuss the evaluation methdology and results.

3.1 Datasets from Twitter and Instagram

In order to evaluate the performance of our algorithm, we collected real world datasets using Twitter and Instagram on *protest* events. We used the query word "protest" to crawl data from both social networks. Twitter provides an API to collect all tweets containing the query word. For Instagram, we used the web service `picodash.com` to collect all the Instagram images containing the query word as an image tag. The data was collected for a period of one month in February 2016, totalling 295, 643 tweets from Twitter and 5, 688 photo posts from Instagram. The weekly statistics of the data are shown in Table 1.

Table 1: Statistics of collected datasets

Week Index	#Tweets	#Instagram posts
Feb 2016 Week 1	77001	1377
Feb 2016 Week 2	78334	1424
Feb 2016 Week 3	75639	1489
Feb 2016 Week 4	64669	1398

3.2 Methodology and Results

We evaluate two aspects of performance of our event tracking system that correspond to our two major contributions; (1) event detection by the fusion module and (2) event tracking, respectively. The baselines in the evaluation are underlying detectors used separately; that is to say, the event detector that exploits Twitter data (called "Twitter") [17], and the detector that finds (Twitter-corroborated) Instagram events (called "Instagram") [5].

3.2.1 *Event Detection by Fusion.* Table 2 summarizes the precision and recall of all three algorithms at event detection during a randomly selected period of 7 days, where we manually labeled all ground truth. More precisely, since we do not know exactly how many events occurred that might have not been reflected in either data set, we abuse recall by referring to the absolute *number of true events* detected. From the table, as expected, we observe that the Instagram algorithm has the highest precision but lowest recall, whereas the Twitter algorithm has the lowest precision.

Table 2: Precision and recall

Algorithm	Total# events	Precision	Recall
Instagram	54	**87.037%**	47
Twitter	174	63.218%	110
Fusion	211	70.616%	**149**

We also investigate the F1 score for all algorithms, as shown in Figure 4. Since, we do not know the ground-truth total number of true events that occurred in each window (to properly compute recall), we plot the F1 score for different values of such total on the x-axis. From Figure 4, we observe that although the Instagram algorithm has the highest precision, its F1 score is consistently the lowest due to its poor recall. In contrast, our fusion-based solution has the highest F1 score, which means that it offers the best trade-off between precision and recall compared to the baselines.

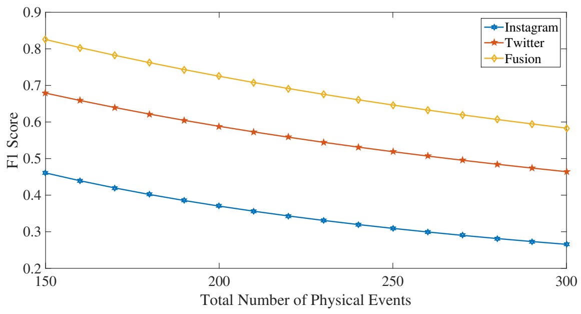

Figure 4: F1 score comparison with varied ground truth of total number of events

Next, we study the performance of demultiplexing (i.e., proper separation of different event instances). Ideal demultiplexing requires that an event detector output a separate event bucket for each event instance. Hence, the number of different events mentioned in tweets in any one bucket should be exactly 1. Accordingly, we use the average number of events (mentioned) per bucket as the metric to evaluate quality of demultiplexing.

We also prefer an event detector that outputs only one bucket for each true physical event. Such a detector has no redundant detections. We use the metric of the number of buckets per detected true event to evaluate redundancy. Note that, for both metrics, the closer to 1 the better in demultiplexing quality and detection redundancy.

Table 3: Demultiplexing quality and detection redundancy

Metric	Instagram	Twitter	Fusion
#events per bucket	**1.078**	1.290	1.194
#buckets per event	1.28	1.35	**1.24**

The results are summarized in Table 3. It shows that our solution has the best performance in terms of reducing redundancy. The intuition behind it is that our solution fuses I-buckets and T-buckets. It is possible that two T-buckets corresponding to the same physical event are correlated with the same I-bucket. They would thus be merged by our fusion algorithm thereby reducing redundancy in event detection. Same argument goes for the I-buckets as well. The Instagram approach remains the best at demultiplexing quality. This is attributed to the location-centric nature of Instagram clusters. A cluster of images from the same location is more likely to describe a single event.

3.2.2 *Event Tracking.* For event tracking, we empirically observed that running the tracking algorithm on top of the I-buckets alone does not consolidate images of the same event over different windows. This might be because image tags on Instagram are chosen by users in a more independent fashion than the wording of tweets on Twitter. In order to observe the evolution of some event instance, we thus cannot exploit Instagram in isolation. Rather, we must also exploit Twitter.

In this section, we use a case study to showcase the performance of our event tracking. Table 4 shows a segment from a tracked event of Delhi protest that sabotaged water supplies for more than 10 million people. In the beginning of this event, people tended to tweet more about the fact, that is Delhi water supplies were sabotaged. The protest later became violent, and India sent soldiers to the area of protest. Next, the impact of this protest was estimated (that more than 10 million people in India were without water), and finally the protest terminated. There were also after-effects that we truncated due to page limits. Without the tracking capability, we could not be able to automattically stitch together the progression of this event. Table 4 also shows one I-bucket for this event and it was fused with a corresponding T-bucket. The case study demonstrates the effectiveness of our tracking solution and the feasibility of automatically merging posts from Twitter and Instagram about the same events.

4 RELATED WORK

Event detection is an extremely popular research topic in the social network community. Much prior to the rise of social media, detection of objects with physical sensors has been studied extensively in the sensor network community [7, 13, 19]. The social sensing field in particular tries to tackle problems related to detection, localization, and tracking the events that occur in a physical space over a period of time. In the following subsections we briefly describe the contributions made in this field using the two popular social networks.

4.1 Event Detection using Twitter

Event detection is one category that has been widely explored with the help of *Twitter*. Topic modeling is a common approach for event detection [10, 12, 20]. Lau et al. [12] proposed an online variation of Latent Dirichlet Allocation (LDA) [4]. In LDA, each topic is modeled as a multinomial distribution of words in a volcabulary, and each document is modeled as a multinomial distribution of k topics, where k is a predifined parameter denoting the total amount of topics. And these two classes of multinomial distributions have two Dirichlet priors respectively. The idea in Lau et al. [12] is incrementally updating the priors in each time window based on the previous calculated parameters, and maintaining the one-to-one correpondence of the topics in the current time window and the last one. If there is a sudden change in the topic word distribution, then a new event is supposed to be detected, where the distance of the distributions is measured by the Jensen-Shannon divergence. Hu et al. [10] proposed ET-LDA (joint Event and Tweets LDA) that exploits a search engine aligns tweets with the corresponding texts of events provided by traditional media, and they showed the results greatly improved. Zhou et al. [20] further expand LDA with time and location of the tweets, and proposed a new graphical model called location-time constrained topic (LTT). In their approach, the tweet content, timestamps and geo-tags are all considered. However, the topic modeling-based approaches usually suffer in the senario that multiple event instances happen in parallel, even when they exploit meta data of the tweets like timestamps and geo-tags [3].

There have also been a few works in determining the reliability of the texts as well as the sources posting information on Twitter. In [15, 16] the authors have focused on the data reliability issue to find the true information from the noisy crowd data. The benefits of using humans directly as sensors include the capability of sensing information in high semantics in real time, which is not possible for physical sensors. However, due to the freedom of posting (almost) any content on social networks, the crowd data is usually very noisy containing rumors, partial information, or polarized viewpoints, which introduces the data reliability issue in social sensing. Wang et al. [15, 16] and a more recent work [18] proposed variant EM based algorithms to address the data reliability challenge by jointly estimating the data authenticity and source reliability. Tracking of events using Twitter was recently explored in [17] where real world datasets have been analyzed to find the evolution of subevents in time and space.

4.2 Event Detection using Instagram

Instagram has emerged as a popular platform among researchers to analyze social networks from a crowdsensing point of view due to an explosive growth in the number of users. In [8], the authors have conducted a study to use Instagram as a social media visualization tool to identify cultural dynamics in major cities. The study particularly zoomed into the city of Tel Aviv, Israel, for a period of two weeks collecting images shared on important national event days. In [11], an analysis was presented to identify different types of users on Instagram and the categories of pictures they take. The work characterized Instagram based on eight categories of pictures shared by five distinct types of users. Prior work [14] also described an approach capable of identifying important *tourist attractions* (POIs) with the help of Instagram. The focus of that work was to identify locations that are extensively visited by tourists. The authors of [9] described the implementation of a system capable of detecting events using geo-tagged data from networks that include Instagram. Their method determines a burst of keywords (tags) within a time interval, which is then modeled by Gaussians, and events are detected based on mapping the bursts. A very recent work [6] explores the techniques to detect and localize events in urban spaces. This work proposes an algorithm that focuses on using the distribution properties of the pictures related to an event in the time domain along with geo-coordinates to do an adaptive clustering followed by false positive elimination. There are a few other event detection techniques using Instagram but to our best knowledge no work has been done to track evolving events.

Contrary to all the related work, in this paper we try to demonstrate the capability of jointly using Twitter and Instagram to not only detect events but also develop a deeper understanding on how these events evolve over a period of time. The combination of data from two different social networks results in better corroboration thus giving a higher precision over the baseline techniques.

5 CONCLUSIONS

We proposed an online event tracking system that integrates Twitter and Instagram data using an unsupervised approach that does not rely on language-specific features. Real-world data evaluation

Table 4: Segment of a tracked event instance of Delhi protest

Window ID	Tweets Sample	Bucket Type
1	VIDEO: Delhi water supplies sabotaged by protest	T-bucket
	No water left in Delhi due to Jat protest, schools closed, rationing begins due to CASTE…	
2	Water rationed as India caste protest toll rises	T-bucket
3	Caste violence …violent protest had briefly shut down the water supply in New Delhi.	T-bucket
	…that Jat started protest, they destroyed munak canal, no water for Delhi	
4	Heaven help us all. Upper caste protest in Delhi leads to death and destruction…	T-bucket
5	India Sends Soldiers To Area Of Caste Protest, water cut off in New Delhi	T-bucket
	Caste Protests Near Delhi Close Roads and Restrict #Water Supply: Though the Indian Army	
6	RT @fakingnews: Fed up Delhi youth start a protest against protests	T-bucket
7	More than 10 million people in #India's capital are without water despite the army regaining control of its key water source after protest	T-bucket
8	Haryana State in India Proposes New Caste Status in Bid to Quell Protests	Fusion bucket
	Jats protest leaves millions in New Delhi without water	
9	Deal reached to end Jat protests in India's Haryana state; roadblocks to be cleared, protest leader and police say	T-bucket

results demonstrate the effectiveness of the proposed system in event detection and tracking. Specifically, compared with two state-of-the-art baselines, our solution offers a better trade-off between precision and recall, lower instances of redundant event detection, and better monitoring of event evolution over time.

REFERENCES

[1] Instagram statistics. http://expandedramblings.com/index.php/important-instagram-stats/.

[2] Twitter statistics. https://www.omnicoreagency.com/twitter-statistics/.

[3] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, and A. Jaimes. Sensing trending topics in twitter. *Multimedia, IEEE Transactions on*, 15(6):1268–1282, 2013.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[5] P. Giridhar, T. Abdelzaher, and L. Kaplan. Social fusion: Integrating twitter and instagram for event monitoring. In *UIUC tech report*, 2017.

[6] 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.

[7] C. Gui and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *MobiCom*. ACM, 2004.

[8] N. Hochman and L. Manovich. Zooming into an instagram city: Reading the local through social media. *First Monday*, 18(7), 2013.

[9] P. Houdyer, A. Zimmerman, M. Kaytoue, M. Plantevit, J. Mitchell, and C. Robardet. Gazouille: Detecting and illustrating local events from geolocalized social media streams. In *Machine Learning and Knowledge Discovery in Databases*, pages 276–280. Springer, 2015.

[10] Y. Hu, A. John, D. D. Seligmann, and F. Wang. What were the tweets about? topical associations between public events and twitter feeds. In *ICWSM*, 2012.

[11] Y. Hu, L. Manikonda, S. Kambhampati, et al. What we instagram: A first analysis of instagram photo content and user types. *Proceedings of ICWSM. AAAI*, 2014.

[12] J. H. Lau, N. Collier, and T. Baldwin. On-line trend analysis with topic models:\# twitter trends detection topic model online. In *COLING*, pages 1519–1534, 2012.

[13] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng. Efficient in-network moving object tracking in wireless sensor networks. *IEEE TMC*, 5(8):1044–1056, 2006.

[14] T. Silva, P. de Melo, J. Almeida, J. Salles, and A. Loureiro. A picture of instagram is worth more than a thousand words: Workload characterization and application. In *2013 IEEE DCOSS*, pages 123–132, May 2013.

[15] D. Wang, T. Amin, S. Li, T. A. L. Kaplan, S. G. C. Pan, H. Liu, C. Aggrawal, R. Ganti, X. Wang, P. Mohapatra, B. Szymanski, and H. Le. Humans as sensors: An estimation theoretic perspective. In *IPSN*, 2014.

[16] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher. On truth discovery in social sensing: a maximum likelihood estimation approach. In *IPSN*, 2012.

[17] S. Wang, P. Giridhar, H. Wang, L. Kaplan, T. Pham, A. Yener, and T. Abdelzaher. Storyline: On physical event demultiplexing and tracking in social spaces. In *IoTDI*, 2017.

[18] S. Wang, L. Su, S. Li, S. Hu, T. Amin, H. Wang, S. Yao, L. Kaplan, and T. Abdelzaher. Scalable social sensing of interdependent phenomena. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pages 202–213. ACM, 2015.

[19] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM Sigmod Record*, 31(3):9–18, 2002.

[20] X. Zhou and L. Chen. Event detection over twitter social media streams. *The VLDB journal*, 23(3):381–400, 2014.