

Detecting Latest Local Events from Geotagged Tweet Streams

Hong Wei¹, Hao Zhou¹, Jagan Sankaranarayanan¹, Sudipta Sengupta² and Hanan Samet¹

¹Department of Computer Science, University of Maryland, College Park, MD 20742

²Amazon Web Services (AWS), Seattle, WA 98101

¹{hyw,hzhou,jagan,hjs}@cs.umd.edu ²sudipta@amazon.com

ABSTRACT

Geotagged tweet streams contain invaluable information about the real-world local events like sports games, protests and traffic accidents. Timely detecting and extracting such events has various applications but yet unsolved challenges. In this paper, we present DELLE, a methodology for automatically **D**etecting **L**atest **L**ocal **E**vents from geotagged tweet streams. DELLE first finds unusual locations which have aggregated unexpected number of tweets, and then ranks the unusual locations to select the top ones that are likely to be local event candidates. We evaluate DELLE on the city of Seattle, WA as well as a larger city of New York. The results show that the proposed method generally outperforms competitive baseline approaches.

CCS CONCEPTS

•Information systems → Data streaming; Clustering;

KEYWORDS

Twitter, Geotagged Tweet Stream, Local Events, Event Detection

ACM Reference format:

Hong Wei, Hao Zhou, Jagan Sankaranarayanan, Sudipta Sengupta and Hanan Samet. 2018. Detecting Latest Local Events from Geotagged Tweet Streams. In *Proceedings of 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle, WA, USA, November 6–9, 2018 (SIGSPATIAL '18), 4 pages.

1 INTRODUCTION

With people posting what is happening outside in the real world, tweets in Twitter encapsulate invaluable information on real-world events as they break. Geotagged tweets are particularly interesting in the sense that they provide the complement information about the place of interest, e.g., where the events occur. In this paper, we aim to detect the latest local events from live geotagged tweet streams. A local event is defined as an unusual activity that appears at some specific time and place and also shows topical coherence. For instance, Figure 1 presents some examples of geotagged tweets about a soccer game held in the city of Seattle, WA. Timely discovering such local events has a wide range of applications. For example, people can acquire the latest information about local activities in their living town., thereby enhancing their daily lives. It can also be helpful for commuting alarms by reporting real-time traffic jams



Figure 1: Examples of geotagged tweets about the soccer game of “Seattle Sounders” vs “D.C. United” at the stadium of “CenturyLink Field” at 7:30 PM, 2017-07-19. All the tweets were located at the stadium of “CenturyLink Field”.

or accidents. In such cases, after learning what is happening, commuters can actively make a decision to bypass the congested road segments or avoid the accident sites.

It is, however, challenging to detect local events from live geotagged tweet streams. For example, detecting local events by capturing unusualness requires considering not only temporal historical patterns but also spatial circumstances. Some studies [1–4] measure the burstiness, intensity of increment in the number of tweets at a place over a short time period, as signals of local events. But burstiness does not always imply the occurrence of a local event. For example, the burstiness of tweets at a shopping mall or a famous coffee bar in the morning is often expected and not unusual. Some work improves this measure to capture temporally routine patterns by gathering time-aware statistics [5]. However, without geographical consideration, occasional nation-wide events may also accumulate a temporally unusual number of tweets at local places. For example, on the presidential election night, one may suddenly observe more tweets everywhere.

In this paper, we propose DELLE to discover local events from live geotagged tweets. The contribution of DELLE lies in its two modules: *seeker* and *ranker*. Seeker finds unusual locations which exhibit spatiotemporal unusualness with respect to the number of tweets and therefore potentially correspond to local events. For this purpose, seeker employs a novel prediction-based anomaly detection strategy. In particular, seeker first exploits convolutional LSTMs (ConvLSTM [6]) to predict the expected number of tweets in the future, which accounts not only for historical patterns but also for neighboring locations. Next, seeker compares the predicted value with the actual number of tweets to determine the existence of unusualness. Unlike previous studies [2, 7] which claim anomalies only based on the local time series data of a location, we also consider the horizontal situation in other places simultaneously to mitigate the effects of global events.

Ranker suppresses the possibly noisy candidates of local events. In practice, not all spatiotemporal burstinesses necessarily correspond to an actual local event. We therefore bring order to the candidates with a ranking procedure by considering temporal burstiness, spatial burstiness and topical coherence, and thereby select the top ones likely to be corresponding to the occurrence of local events.

2 RELATED WORK

There has been a lot of works on detecting local events using tweets in Twitter. Atefeh [8] and Abdelhaq [9] provide two excellent surveys. In general, existing methods focusing on geotagged tweets can be classified into two strategies: *model dimension extension* and *geographical space tessellation*. Model dimension extension treats location as additional factors to existing models, e.g., latent variables in the generative topic model [10–13].

Geographical space tessellation divides space into small and disjoint cells for aggregating geotagged tweets. The motivation is that a local event usually has a limited spatial impact and would fall in the same or nearby cell(s). The grid is the simplest yet most commonly used way of tessellation [2, 14–17], although other structures have also been explored including hierarchical triangular meshes [7].

After aggregating tweets to tessellation cells, a simple way for event detection is to examine whether the number of the aggregated tweets or the arriving rate exceeds a certain threshold [1, 5]. This, however, is easily plagued by tweet distribution heterogeneity both temporally and spatially. Various anomaly detection methods are then explored. The core idea is to use history data to build a baseline (or make a prediction) and then compare with the actual value to check for significant discrepancies [2, 3, 7]. For example, TwitInfo [18] uses the weighted average of historical tweet counts to compute the expected frequency of tweets. But sole historical data often neglect the effects exerted by nearby geographical regions. Krumm and Horvitz [7] therefore include features like tweet counts from adjacent regions in their anomaly detection method. Our method is different from the above methods in two senses: 1) our prediction model captures both spatial dependencies and temporal patterns [19]; 2) when claiming an anomaly, we account not only for the history of a location itself but also for the situation at other places to mitigate the effect of unexpected global events.

The most related work to our task are EVENTWEET [2], Eyewitness [7]. EVENTWEET [2] detect events by identifying and clustering temporal bursty keywords. However, using words instead of tweets as clustering elements, this method may group semantically irrelevant words together. Eyewitness [7] discretizes space and time and finds tweet volume spikes as potential local events by comparing the predicted value with the observed value. However, it needs to perform an exhaustive sweep through different space and time pieces and thereby is not easy to modify for online processing.

Due to the sparsity of geotagged tweets (1%), some methods try to acquire more local tweets by tracking local people [20, 21] or utilizing geotagging procedures [22–25] for location estimation. The location information in these methods, however, are usually in a very coarse resolution (e.g., city-level) and rarely used when grouping tweets together. Some methods try to first detect an event and estimate its location afterwards, e.g., TwitterStand [26–30]. These methods are different from our focus as we instead try to extract local events from geotagged tweet streams.

3 METHOD

3.1 Problem

Given a geotagged tweet stream, our goal is to identify the latest local events. Formally, suppose that t is the current (latest) time point and Δt is a short time interval, we define $\mathcal{D}_t$ to be the geotagged tweet stream up to t , and $\mathcal{D}_{t-\Delta t \rightarrow t}$ be the geotagged tweet stream from $t - \Delta t$ to t . In other words, $\mathcal{D}_{t-\Delta t \rightarrow t}$ essentially represents the latest geotagged tweets with respect to Δt . For simplicity, a geotagged tweet d can be seen as a tuple $\langle time_d, loc_d, txt_d, user_d \rangle$ in which $time_d$ is the publication time, loc_d is the geographical location (i.e., a pair of lat/long coordinates), txt_d refers to the textual content and $user_d$ is the user posting this tweet. The latest local event detection problem is then to extract from $\mathcal{D}_{t-\Delta t \rightarrow t}$ all possible local events, where each event is a cluster of geographically, temporally and semantically close tweets.

3.2 System

In order to detect the local events from $\mathcal{D}_{t-\Delta t \rightarrow t}$, we first discretize the geotagged tweet stream into a set of disjoint intervals, i.e., $\{ \dots [t - 2\Delta t, t - \Delta t), [t - \Delta t, t) \}$. We then utilize a uniform grid to tessellate the spatial region into squares of size $\Delta l \times \Delta l$, where Δl is the side length of the square. After discretizing space, the tweets are subsequently fed into a pipeline of two modules: *seeker* and *ranker*. *Seeker* finds spatiotemporal unusualness in the number of tweets as potential candidates of local events. *Ranker* selects which set of unusualness are most likely to be local events.

3.2.1 Seeker. After tessellating the space into an $M \times N$ grid and discretizing time into periods of length Δt , the task of seeker is to identify grid cells that show an unusual aggregation of tweets in latest geotagged tweet stream $\mathcal{D}_{t-\Delta t \rightarrow t}$ or $\mathcal{D}_T$ where T denotes the last time interval of length Δt .

Tweet Count Prediction The goal of tweet count prediction is to use previously historical tweet count data in a local region to forecast on the number of tweets to appear in the next time step [19]. On an $M \times N$ grid map, the tweet count values in the grid cells at time step τ can be written in a tensor $\mathbf{X}_\tau \in \mathbb{R}^{M \times N}$ where $X_\tau(m, n)$ is the tweet count in the grid cell (m, n) at time step τ . Therefore, the prediction problem is formulated as follows:

Definition 3.1. The tweet count prediction problem $\mathcal{P}$ is to generate a prediction $\mathbf{Y}_T$, which is an estimation of $\mathbf{X}_T$, given a list of historical observations $\{\mathbf{X}_\tau | \tau = 0, \dots, T - 1\}$.

In this paper, we utilize a residual Convolutional LSTM (ConvLSTM [6]) based prediction model [19], which is reported to have state-of-the-art accuracy. The prediction model in [19] is made up of three main branches: *closeness*, *period* and *trend*, to incorporate temporal pattern information at a different scale in tweet data, together with a meta-data branch to capture features such as time-of-day (e.g., in minutes), day-of-week.

From Prediction To Unusualness We define the prediction error to be $E_T = Y_T - X_T$, where X_T is the latest tweet count on a spatial $M \times N$ grid and Y_T is the prediction of X_T . $E_T(m, n)$ indicates the prediction error of the grid cell (m, n) . Intuitively, a significant negative $E_T(m, n)$ indicates a local event as there were many more tweets than usual. Following [7], we define the precision of our prediction model to be σ_{E_T} , where $\sigma_{E_T}(m, n)$ is the standard deviation of the grid cell (m, n) w.r.t. its history of prediction errors

$\mathcal{E}_T(m, n) = \{\dots \mathcal{E}_{T-1}(m, n), \mathcal{E}_T\}$. To account for the precision of the prediction model, we re-define the prediction error as:

$$\mathcal{E}'_T = \mathcal{E}_T \oslash \sigma_{\mathcal{E}_T} \quad (1)$$

where $\oslash$ denotes the element-wise division operation.

To detect unusual grid cells using $\mathcal{E}'_T$, we utilize an image restoration framework called Deep Image Prior [31]. Our intuition is that the unusualness in $\mathcal{E}'_T$ is like spike noise in an image, and Deep Image Prior can be used to denoise corrupted images without prior knowledge of training data. Suppose that $\mathcal{E}''_T$ is the restored image of $\mathcal{E}'_T$, and $\Delta \mathcal{E}'_T = \mathcal{E}''_T - \mathcal{E}'_T$, we claim a grid cell (m, n) is unusual if

$$|\Delta \mathcal{E}'_T(m, n) - \mu_{\Delta \mathcal{E}'_T}| \geq k_{\Delta \mathcal{E}'_T} \cdot \sigma_{\Delta \mathcal{E}'_T} \quad (2)$$

where $\mu_{\Delta \mathcal{E}'_T}$ and $\sigma_{\Delta \mathcal{E}'_T}$ are the mean and standard deviation of grid cells in $\Delta \mathcal{E}'_T$, respectively. $k_{\Delta \mathcal{E}'_T}$ is a predefined threshold for determining the unusualness of a grid cell. Different from [7], our approach accounts for both history of a grid cell and information of other cells on the whole region when detecting unusualness in a location. This is important in differentiating global events which might cause an unusual number of tweets on a local grid cell.

3.2.2 Ranker. In this section, we make a ranking of these unusual locations to identify the top ones that are most likely corresponding to the occurrence of local events, by addressing temporal burstiness, spatial burstiness and topical coherence.

Temporal Burstiness For a grid cell (m, n) , suppose that $\mathcal{Y}_T(m, n)$ represents a history of estimations on its number of tweets up to the time step T , and is defined as:

$$\mathcal{Y}_T(m, n) = \{\dots \mathcal{Y}_{T-1}(m, n), \mathcal{Y}_T(m, n)\} \quad (3)$$

Then we use z-score to quantify the grid cell (m, n) 's temporal burstiness [32] at T , denoted as $TB_T(m, n)$ and defined as:

$$TB_T(m, n) = \frac{X_T(m, n) - \mu_{\mathcal{Y}_T(m, n)}}{\sigma_{\mathcal{Y}_T(m, n)}} \quad (4)$$

where $\mu_{\mathcal{Y}_T(m, n)}$ and $\sigma_{\mathcal{Y}_T(m, n)}$ are the mean and standard deviation of $\mathcal{Y}_T(m, n)$, respectively. Recall that $X_T(m, n)$ is the actual number of tweets in grid cell (m, n) at time step T .

Spatial Burstiness Given a grid cell (m, n) , the spatial burstiness is measured by the spatial density distribution of keywords of the tweets in (m, n) . The intuition is that a low spatial density distribution means that the keyword is widely spread over space and a high distribution means that the keyword occurs only at a few locations. Therefore, the keywords in local events should have higher spatial density distribution to be spatially bursty.

Suppose that $\mathcal{D}_T(m, n)$ is the tweet set in grid cell (m, n) at T , and $\mathcal{W}_T(m, n)$ is the set of keywords (e.g., after removing stop words) in (m, n) , i.e., $\mathcal{W}_T(m, n) = \{w \mid w \in \text{txt}_d \text{ and } d \in \mathcal{D}_T(m, n)\}$. Let $SDD_T^w(m, n)$ be the spatial density distribution of keyword w in grid cell (m, n) at T , i.e.,

$$SDD_T^w(m, n) = \frac{\# \text{ of } w \text{ in grid cell } (m, n)}{\sum_{(m', n') \in M \times N} \# \text{ of } w \text{ in grid cell } (m', n')} \quad (5)$$

We now define the spatial burstiness of grid cell (m, n) as:

$$SB_T(m, n) = \sum_{w \in \mathcal{W}_T(m, n)} SDD_T^w(m, n) \quad (6)$$

Topical Coherence The topical coherence captures the semantic similarity of tweets in a grid cell. In other words, the tweets posted on the same event should be discussing similar content and probably using similar vocabularies. Tweep2Vec [33] learns

the vector-space representations of tweets using a character-based bi-directional recurrent neural network model, and has been demonstrated to have good performance in the application of clustering semantically similar tweets [34]. To measure the topical similarity between tweets, we use Tweep2Vec to encode a textual tweet in character sequence to a vector embedding with a default dimension size of 500.

Let $TS(d', d'')$ be the topical similarity between tweets d' and d'' . To measure the topical coherence of the tweets in cell (m, n) , we construct a graph, called *Tweet Influence Graph*.

Definition 3.2. (Tweet Influence Graph). The tweet influence graph on the grid cell (m, n) at T , is an undirected graph $G_T = (V_T, E_T)$ where V_T is the set of all tweets in $\mathcal{D}_T(m, n)$, E_T is the set of edges between tweets, and the weight of an edge between d' and d'' is their topical similarity $TS(d', d'')$.

We now employ PageRank [35], a random walk procedure, on the tweet influence graph to bring order to the influence of tweets in $\mathcal{D}_T(m, n)$ and thus identify the top k tweets with the most influence, denote by $\mathcal{D}_T^k(m, n)$. The *topical coherence* is thus defined as:

$$TC_T(m, n) = \frac{\sum_{d' \in \mathcal{D}_T^k(m, n), d'' \in \mathcal{D}_T^k(m, n)} TS(d', d'')}{k^2} \quad (7)$$

The rationale is that if the tweets in $\mathcal{D}_T(m, n)$ are about the same local event, then the most topically influential tweets should have higher topical similarity between each other. One may point out that such a topical coherence measurement would suppress a grid cell having multiple topically unrelated ongoing events. We argue that such a case is very rare with a fine space and time discretization.

3.2.3 Ranking Function. As the final step, we now define the ranking score of the grid cell (m, n) by aggregating its *temporal burstiness*, *spatial burstiness* and *topical coherence*, after rescaling them to $[0, 1]$ with respect to other grid cells:

$$\mathcal{R}_T(m, n) = TB'_T(m, n) \cdot SB'_T(m, n) \cdot TC'_T(m, n) \quad (8)$$

where $TB'_T(m, n) = (TB_T(m, n) - TB_T^{min}) / (TB_T^{max} - TB_T^{min})$ with TB_T^{max} and TB_T^{min} being the maximum and minimum of topical burstiness among all grid cells at T . Spatial burstiness and topical coherence are rescaled in the same way, receptively. After the ranking, we select the top- K and report them as the local events.

4 EVALUATION

DELLE is implemented in Python and evaluated with on a computer with an Intel Xeon E5 CPU, an Nvidia Quadro P6000 GPU and a 64GB RAM.

4.1 Experimental Settings

4.1.1 Datasets. The evaluation is performed on two sets of geo-tagged tweets collected from 2015-07-09 to 2017-07-23 in two cities: Seattle, WA (SEA) and New York City (NYC) [19]. Their geographical regions are two bounding boxes spanning from [47.579784, -122.373135] to [47.633604, -122.293062] for SEA, and from [40.647984, -74.111093] to [40.853945, -73.837472] for NYC. The total number of tweets after removing spam tweets [19], is 756, 457 and 9, 353, 721, respectively. We take the data from 2017-06-23 to 2017-07-23 for testing and local event detection, and its previous data for training the tweet count prediction model.

4.1.2 *Baseline Approaches.* The two baseline approaches are:

- EVENTTWEET [2] first identifies temporal bursty keywords and spatial local keywords and then clusters them to find local events.
- Eyewitness [7] finds tweet volume spikes in discretized time and space as potential local events by comparing the actual number of tweets with the predicted value using a regression model.

4.1.3 *Parameter Settings.* The major parameters in DELLE are set as follows. For space and time, we set the side length of grid cells $\Delta l = 500\text{m}$ and the length of time interval $\Delta t = 30$ minutes (by dividing a natural integral hour into two intervals) since such values provide fine enough resolution for local event detection as well as yield good performance for tweet count prediction [19]. As a result, we have a 12×12 grid map in SEA and 46×46 in NYC. We set the threshold for determining the unusualness of a grid cell $k_{\Delta E'_T} = 3$, a commonly used value for anomaly detection.

In each time interval, we select at most $K = 5$ unusual grid cells as the local event candidates. Because not every time interval necessarily has K local events happening, we apply a simple heuristic for suppressing the negative candidates. It removes grid cells having too few users (i.e., less than 5). For fairness, we also similarly filter out the event candidates with less than 5 users for the baseline approaches as well in the evaluation.

4.2 Effectiveness

Table 1: Comparison results using Precision, Recall and F-Score.

Method	Seattle, WA				NYC			
	#	P	R	F	#	P	R	F
EVENTTWEET	354	0.391	0.390	0.390	1665	0.146	0.131	0.138
Eyewitness	273	0.769	0.593	0.670	1204	0.614	0.398	0.483
DELLE	269	0.862	0.655	0.745	1128	0.741	0.450	0.560

We evaluate the different local event detection methods using *precision*, *recall* and *f-score*. For precision, we recruited 3 volunteers to individually judge the detected events and collect the results using the strategy of majority votes¹. In lack of groundtruth on the set of events happening in the real world, we build a pseudo groundtruth by assembling a set of distinct true positive local events reported in different methods to calculate the recall and f-score. The comparison results are listed in Table 1. It shows that DELLE outperforms baseline approaches in most cases. In particular, a significant improvement is observed over EVENTTWEET.

5 CONCLUSIONS

In this paper, we presented DELLE for detecting latest local events in geotagged tweet streams. In essence, DELLE first identifies spatiotemporal unusualness using a novel prediction-based anomaly detection approach, and subsequently ranks them to identify potential local events, by addressing both spatiotemporal burstiness and topical coherence. The evaluation results on two selected cities show that DELLE outperforms competitive baselines in most cases, showing the effectiveness of the proposed method.

The human evaluation yields a groundtruth of local events, and therefore enables the exploration of learning to classify spatiotemporal unusualness into true/false local events using features like burstiness and topical coherence. We leave this for our future work.

¹The instructions given to the judges are summarized at <http://www.cs.umd.edu/~hyw/instructions-local-events.txt>

6 ACKNOWLEDGEMENT

We would like to thank Dr. John Krumm and Dr. Jin Li from Microsoft Research for providing supporting funding and the access to tweets of Twitter. This work was also supported in part by the NSF under Grants IIS-13-20791 and IIS-1816889.

REFERENCES

- [1] M. Mathioudakis and N. Koudas. TwitterMonitor: Trend Detection over the Twitter Stream. SIGMOD '10.
- [2] H. Abdelhaq, C. Sengstock, and M. Gertz. EvenTweet: Online Localized Event Detection from Twitter. PVLDB '13.
- [3] T. Lappas, M. R. Vieira, D. Gunopulos, et al. On the Spatiotemporal Burstiness of Terms. PVLDB '12.
- [4] Q. He, K. Chang, and E.-P. Lim. Analyzing Feature Trajectories for Event Detection. SIGIR '07.
- [5] H. Abdelhaq, M. Gertz, and C. Sengstock. Spatio-temporal Characteristics of Bursty Words in Twitter Streams. SIGSPATIAL '13.
- [6] X. Shi, Z. Chen, H. Wang, et al. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. NIPS '15.
- [7] J. Krumm and E. Horvitz. Eyewitness: Identifying Local Events via Space-time Signals in Twitter Feeds. SIGSPATIAL '15.
- [8] F. Atefeh and W. Khreich. A Survey of Techniques for Event Detection in Twitter. *Comput. Intell.*, 31(1):132–164, February 2015.
- [9] H. Abdelhaq. *Localized Events in Social Media Streams: Detection, Tracking, and Recommendation*. PhD thesis, Heidelberg University, November 2015.
- [10] C. Zhang, L. Liu, D. Lei, et al. TrioVecEvent: Embedding-Based Online Local Event Detection in Geo-Tagged Tweet Streams. KDD '17.
- [11] L. Hong, A. Ahmed, S. Gurumurthy, et al. Discovering Geographical Topics in the Twitter Stream. WWW '12.
- [12] X. Zhou and L. Chen. Event Detection over Twitter Social Media Streams. VLDB, 23(3):381–400, June 2014.
- [13] W. Wei, K. Joseph, W. Lo, et al. A Bayesian Graphical Model to Discover Latent Events from Twitter. ICWSM '15.
- [14] A. Skovsgaard, D. Sidlauskas, and C. S. Jensen. Scalable top-k spatio-temporal term querying. ICDE '14.
- [15] K. Y. Kamath, J. Caverlee, K. Lee, et al. Spatio-temporal Dynamics of Online Memes: A Study of Geo-tagged Tweets. WWW '13.
- [16] K. Watanabe, M. Ochi, M. Okabe, et al. Jasmine: A Real-time Local-event Detection System Based on Geolocation Information Propagated to Microblogs. CIKM '11.
- [17] C. Jonathan, A. Magdy, M. F. Mokbel, et al. GARNET: A holistic system approach for trending queries in microblogs. ICDE '16.
- [18] A. Marcus, M. S. Bernstein, O. Badar, et al. Twitinfo: Aggregating and Visualizing Microblogs for Event Exploration. CHI '11.
- [19] H. Wei, H. Zhou, J. Sankaranarayanan, et al. Residual Convolutional LSTM for Tweet Count Prediction. WWW '18 Companion.
- [20] J. Weng and B.-S. Lee. Event Detection in Twitter. ICWSM '11.
- [21] H. Wei, J. Sankaranarayanan, and H. Samet. Finding and Tracking Local Twitter Users for News Detection. SIGSPATIAL '17.
- [22] M. D. Lieberman and H. Samet. Multifaceted Toponym Recognition for Streaming News. SIGIR '11.
- [23] M. D. Lieberman and H. Samet. Adaptive Context Features for Toponym Resolution in Streaming News. SIGIR '12.
- [24] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging: Using Proximity, Sibling, and Prominence Clues to Understand Comma Groups. GIR '10.
- [25] H. Samet. Using Minimaps to Enable Toponym Resolution with an Effective 100% Rate of Recall. GIR '14.
- [26] J. Sankaranarayanan, H. Samet, B. E. Teitler, et al. TwitterStand: News in Tweets. SIGSPATIAL '09.
- [27] N. Gramsky and H. Samet. Seeder Finder: Identifying Additional Needles in the Twitter Haystack. LBSN '13.
- [28] A. Jackoway, H. Samet, and J. Sankaranarayanan. Identification of Live News Events Using Twitter. LBSN '11.
- [29] H. Samet, J. Sankaranarayanan, M. D. Lieberman, et al. Reading News with Maps by Exploiting Spatial Synonyms. *Commun. ACM*, 2014.
- [30] M. D. Lieberman and H. Samet. Supporting Rapid Processing and Interactive Map-based Exploration of Streaming News. SIGSPATIAL '12.
- [31] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Deep Image Prior. CVPR '17.
- [32] C. Zhang, G. Zhou, Q. Yuan, et al. GeoBurst: Real-Time Local Event Detection in Geo-Tagged Tweet Streams. SIGIR '16.
- [33] B. Dhingra, Z. Zhou, D. Fitzpatrick, et al. Tweet2Vec: Character-Based Distributed Representations for Social Media. ACL '16.
- [34] S. Vakulenko, L. Nixon, and M. Lupu. Character-based Neural Embeddings for Tweet Clustering. SocialNLP '17.
- [35] L. Page, S. Brin, R. Motwani, et al. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford InfoLab, 1999.