http://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 34, No. 2, March–April 2022, pp. 790–806
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

SHEDR: An End-to-End Deep Neural Event Detection and Recommendation Framework for Hyperlocal News Using Social Media

Yuheng Hu,[a] Yili Hong[b]

[a] Department of Information and Decision Sciences, University of Illinois at Chicago, Chicago, Illinois 60607; [b] Department of Decision and Information Sciences, University of Houston, Houston, Texas 77004

Contact: yuhenghu@uic.edu, https://orcid.org/0000-0003-3665-1238 (YuH); yilihong@uh.edu (YiH)

Received: May 20, 2019
Revised: August 17, 2020; January 7, 2021; June 6, 2021
Accepted: June 9, 2021
Published Online in Articles in Advance: December 13, 2021

https://doi.org/10.1287/ijoc.2021.1112

Abstract. Residents often rely on newspapers and television to gather hyperlocal news for community awareness and engagement. More recently, social media have emerged as an increasingly important source of hyperlocal news. Thus far, the literature on using social media to create desirable societal benefits, such as civic awareness and engagement, is still in its infancy. One key challenge in this research stream is to timely and accurately distill information from noisy social media data streams to community members. In this work, we develop SHEDR (social media–based hyperlocal event detection and recommendation), an end-to-end neural event detection and recommendation framework with a particular use case for Twitter to facilitate residents' information seeking of hyperlocal events. The key model innovation in SHEDR lies in the design of the hyperlocal event detector and the event recommender. First, we harness the power of two popular deep neural network models, the convolutional neural network (CNN) and long short-term memory (LSTM), in a novel joint CNN-LSTM model to characterize spatiotemporal dependencies for capturing unusualness in a region of interest, which is classified as a hyperlocal event. Next, we develop a neural pairwise ranking algorithm for recommending detected hyperlocal events to residents based on their interests. To alleviate the sparsity issue and improve personalization, our algorithm incorporates several types of contextual information covering topic, social, and geographical proximities. We perform comprehensive evaluations based on two large-scale data sets comprising geotagged tweets covering Seattle and Chicago. We demonstrate the effectiveness of our framework in comparison with several state-of-the-art approaches. We show that our hyperlocal event detection and recommendation models consistently and significantly outperform other approaches in terms of precision, recall, and F-1 scores.

Summary of Contribution: In this paper, we focus on a novel and important, yet largely underexplored application of computing—how to improve civic engagement in local neighborhoods via local news sharing and consumption based on social media feeds. To address this question, we propose two new computational and data-driven methods: (1) a deep learning–based hyperlocal event detection algorithm that scans spatially and temporally to detect hyperlocal events from geotagged Twitter feeds; and (2) A personalized deep learning–based hyperlocal event recommender system that systematically integrates several contextual cues such as topical, geographical, and social proximity to recommend the detected hyperlocal events to potential users. We conduct a series of experiments to examine our proposed models. The outcomes demonstrate that our algorithms are significantly better than the state-of-the-art models and can provide users with more relevant information about the local neighborhoods that they live in, which in turn may boost their community engagement.

History: Accepted by Ram Ramesh, Area Editor for Knowledge Management and Machine Learning.

Keywords: events • event detection • event recommendation • deep learning • civic engagement

1. Introduction

Recent years have witnessed a dramatic decline of neighborliness, civic engagement, and volunteerism in society (Stolle and Hooghe 2005). Famed sociologist Robert D. Putnam characterizes such decline as the result of drops of social capital in contemporary America communities (Putnam 2000). He argues that social capital is vital to the life and health of a community, because "it has many features that help people translate aspirations into realities" (Putnam 2000, p. 228).

According to Putnam, social capital widens the awareness of individuals' mutual connectivity, which ultimately enhances a nation's civic and democratic institutions. Further, social capital can lead to more bipartisan attempts to addressing societal issues which, in turn, can help improve social environments, such as safer neighborhoods.

One way to improve the social capital of local communities is through local information sharing. In fact, accessing information about local communities has been widely regarded as an important indicator of community awareness and engagement (Putnam 2000). At the conceptual level, this perspective resonates with the concept of *hyperlocal news*, which refers to geographically bounded news most relevant to local community members. For example, Metzgar et al. (2011, p. 774) use several criteria to quantify hyperlocal news operations such as "geographically-based, community-oriented, original-news-reporting organizations indigenous to the web and intended to fill perceived gaps in coverage of an issue or region and to promote civic engagement."

For a long time, to access hyperlocal news, local residents have relied on local newspapers, TV, and radio as the primary sources. More recently, social media platforms such as Twitter have emerged as an important tool for hyperlocal news (Carroll et al. 2011). Compared with the aforementioned traditional sources, Twitter has several unique features worth consideration. First, eyewitnesses can report and broadcast hyperlocal news on social media before traditional news outlet can, thanks to social media's ubiquity and immediacy. Thus, residents can obtain important news about their community in a timelier fashion. Moreover, social media enable the seamless integration of consumption, connection, and interaction. Since hyperlocal news is mostly peer-generated by, consumed by, and shared by residents of a local community, social media can effectively create an interactive environment for them to access the hyperlocal news. Despite these benefits, social media data are known to be highly noisy, posing significant challenges to individuals in effectively extracting high-quality information. Therefore, it is no surprise that a majority of local residents still rely on traditional sources such as TV and newspapers as their primary channels for hyperlocal information (Tezon 2003).

The primary goal of this paper is to develop a computational framework for extracting relevant, high-quality hyperlocal content from a noisy Twitter stream to facilitate resident's information-seeking of hyperlocal news, which can potentially improve their civic awareness and engagement. In particular, we focus on *hyperlocal events*, that is, events that happen in local communities. There are several motivations behind our consideration. First, various prior research has shown that hyperlocal events are the most important and sought-after information in hyperlocal news (e.g., Xia et al. 2014). For instance, through a large-scale study, Farnham et al. (2015) categorized the tweets' topics that mention local communities into 14 categories and found that the most popular category is about a current event or happening in the neighborhood (corresponding to 29% of tweets that they analyzed). On the other hand, citizen journalism, defined as public citizens who actively report, collect, analyze, and disseminate news information, has become the main driving force in hyperlocal news production (Bowman and Willis 2003). Compared with traditional professional journalism, citizen journalists mostly focus on news about daily life such as events happening in the community (Bruns et al. 2008). Finally, local residents frequently interact with others about local events in terms of reporting and disseminating information about these events on Twitter (Masden et al. 2014). In sum, hyperlocal events matter to local community members and citizen journalists significantly.

Technically, it is very challenging to detect and recommend hyperlocal news to residents for several reasons. First, detecting hyperlocal events requires considering not only temporal patterns but also spatial circumstances—two important dimensions that characterize a hyperlocal event (Lee and Sumiya 2010). Previous studies tried to detect local events on Twitter by simple measures such as the burstiness in the number of tweets at a place during a short time period (Lee and Sumiya 2010, Zhang et al. 2016). However, burstiness does not sufficiently indicate a hyperlocal event. Some instances of burstiness are expected routine patterns, such as a large number of tweets posted near a coffee shop every early morning. In addition, without considering the geographical aspect, nationwide events (e.g., a Super Bowl game) may also cause unusual tweet volume all over local places occasionally. Finally, the sheer number of available hyperlocal events, especially in local communities with dense populations, may cause the information overload, thus undermining local residents' capability of accessing the events that they are most interested in.

In this work, we propose SHEDR: a social media–based hyperlocal event detection and recommendation framework. Our framework has three main components, namely, a tweet fetcher, a hyperlocal event detector, and a hyperlocal event recommender. First, the fetcher tessellates a region of interest by dividing space into small and disjoint cells in terms of hierarchical triangular mesh for retrieving and aggregating geotagged tweets. After that, the detector scans over those cells and finds ones that exhibit spatiotemporal unusualness with respect to the number of geotagged tweets and considers them prospective hyperlocal events. Finally, the event recommender brings order to those events with a ranking procedure catering to a potential user's own interest.

Our key technical contributions in this work lie in the design of the *event detector* and the *event recommender*. First, we harness the power of two popular deep neural network models, the convolutional neural network (CNN) and long short-term memory (LSTM), to build a novel joint CNN-LSTM model to capture spatiotemporal dependencies for predicting the expected tweet volume from a region of interests at a given time. The predicted value is then compared with the actual volume of tweets to determine if the actual volume is unusual, indicating that a spike exists. Whereas various CNN-LSTM architectures have been proposed and applied to some computer vision and natural language processing problems (Donahue et al. 2015, Wang et al. 2016), to the best of our knowledge, we are the first work in the literature to develop a CNN-LSTM model for detecting hyperlocal events. More importantly, previous CNN-LSTM models rely on full CNN models. However, a full CNN cannot model spatial correlation of entire regions in our scenario. Instead, we propose a localized CNN architecture to characterize the spatial influence of nearby neighbors to the target region of interest, given the first law of geography, that is, "near things are more related than distant things" (Tobler 1970, p. 236). Also, unlike previous models, which focused squarely on detecting events based on anomalies in temporal patterns of tweets of a location (Krumm and Horvitz 2015, Wei et al. 2019), our algorithm also considers the novelty of the events to mitigate the effects of global events or routine events. Next, we develop a clustering-classification algorithm to group the unusualness into event clusters and classify if a spatiotemporal unusualness is indeed a hyperlocal event. Finally, in the event recommender component, we develop a novel deep learning–based pairwise ranking model. To improve personalization, our model incorporates several types of contextual information, including the social proximity derived from the user's and the hyperlocal event participants' Twitter profiles, the geographical proximity according to the locations of the user and the event, and topical proximity based on the matching of the user's interest topics and the event's topics.

We perform comprehensive evaluations based on two large-scale data sets of geotagged tweets covering areas of Seattle and Chicago. We demonstrate the effectiveness of our framework in comparison with several state-of-the-art approaches. We show that our hyperlocal event detection and recommendation algorithms consistently and significantly outperform other approaches in terms of precision, recall, and F-1 scores.

2. Background

2.1. Hyperlocal News and Citizen Journalism

Hyperlocal news refers to geographically bounded news that is most relevant to local communities. Metzgar et al. (2011, p. 774) use six criteria to quantify hyperlocal news operations such as "geographically-based, community-oriented, original-news-reporting organizations indigenous to the web and intended to fill perceived gaps in coverage of an issue or region and to promote civic engagement." Traditionally, professional journalists are the main producer of hyperlocal news. Most recently, citizen journalism, defined as public citizens who actively report, collect, analyze, and disseminate news information, has become the main driving force in hyperlocal news production (Bruns et al. 2008). There are several differences between professional journalism and citizen journalism in hyperlocal news. For example, based on a case study of the Belgian newspaper *Het Belang van Limburg*'s hyperlocal news project, D'heer and Paulussen (2013) found that, while factual news such as crimes is mainly produced by professional journalists, citizen journalists often report soft news about community events.

2.2. Technologies for Facilitating Hyperlocal News and Citizen Journalism

With the growing popularity of online communities and social networking platforms such as Twitter and Facebook, there has been a growing interest in understanding how technology strengthens community image, enhances residents' awareness, and promotes civic engagement (López and Farzan 2015).

The emergence of social media has generated a profound impact on various organizations and society and, recently, has led to research efforts on utilizing social media for the purpose of community awareness and participation (Xia et al. 2014). Due to social media being naturally noisy, Virtual Town Square and Whoo.ly have developed various methods to extract relevant and meaningful information from tweets to improve community engagement (Han et al. 2014, Hu et al. 2013). Such social media–enabled tools have shown a positive impact on civic engagement and community awareness (Farnham et al. 2015).

Our work builds upon and advances the objectives of the extant stream of literature. In particular, we focus on hyperlocal events, as they represent the most important and popular hyperlocal content. We develop an end-to-end deep learning–based hyperlocal event and recommendation framework on Twitter to facilitate a local resident's information seeking of hyperlocal events and, in turn, enhance her or his engagement within the local community. Our key technical contributions in this work include a novel hyperlocal event detection component and a novel event recommendation component that can be readily applied to social media data. We next discuss related prior work.

2.3. Event Detection and Recommendation from Social Media

Automated event detection from social media has been a popular research topic in the technical research communities. In general, there are two main approaches: unspecified and specified event detection. An *unspecified event* refers to an event without prior information (e.g., when and where the event will happen). A typical approach detects an event via monitoring temporal signals, such as a sudden increase in the use of specific words (e.g., *gunshots*) in a social media stream. After that, these words will be grouped into events and classified into different event categories (Atefeh and Khreich 2015). Conversely, a *specified event* means an event for which prior information, such as venue, time, and description, is given. Thus, specified event detection approaches essentially detect an event by extracting the event's related information to augment the event (Becker et al. 2012).

Notably, whereas tweets that share a common topic within a period often represent an event, such an event is not necessarily a "local" event. Thus, existing event detection approaches, regardless of being a specified or unspecified approach, may not be suitable for detecting local events. In recent years, there has been an emerging stream of works on detecting local events on social media. Two popular approaches are typically used to detect an event. The first one is through examining the tweets' content. For example, EvenTweet detects events based on temporal bursty keywords (Abdelhaq et al. 2013). Similarly, GeoBurst uses a prespecified circle and computes the coherence of tweets originating within the circle to form local event clusters (Zhang et al. 2016). The second approach is based on anomaly detection. For example, Eyewitness and DeLLe consider tweet volume spikes as potential local events by comparing the predicted values with the observed values (Krumm and Horvitz 2015, Wei et al. 2019).

Compared with these approaches, our approach has several novel aspects. First, text processing on tweet content is a known challenge task due to Twitter's short format and irregular linguistics. Our approach does not rely on the text content of tweets to detect hyperlocal events, which avoids expansive processing of text data; instead, we use text only for clustering and classification in a postprocessing step to filter out the event. We later show that our model can significantly outperform those text content-based approaches. Second, compared with traditional anomaly detection-based approaches, our joint CNN-LSTM event detection model does not scan the entire region of interests; rather, it scans just a few regions based on temporal dependency via LSTM and spatial dependency via a localized CNN. Thus, our model can reduce unnecessary noise and, therefore, generate more accurate predictions. Plus, our approach can detect multiple local events from a region in a principled manner, whereas the aforementioned approaches can detect only one single event. Thus, our approach is more robust and provides more practical value in identifying hyperlocal events.

On the other hand, there is also a rich body of literature on recommender systems. Traditional recommendation tasks focus primarily on recommending items such as products, points of interest, or people. Compared with traditional recommendation tasks, event recommendation, let alone hyperlocal event recommendation, has received much less attention. The event recommendation problem is challenging, because events are short-lived and typically not known in advance. Both content-based techniques and collaborative filtering techniques have been proposed (Liu et al. 2012). Our work differs from prior research by focusing on recommending hyperlocal events. Further, our recommendation framework is built on a deep learning–based pairwise ranking model, which extends the generalized matrix factorization, thus providing both linearity and nonlinearity for much better expressiveness in the recommendation tasks. Further, our algorithm exploits a wide range of contextual data related to local community and local events such as user profile, geographical event information, as well as social connections between the focal user and event participants to enhance the model performance.

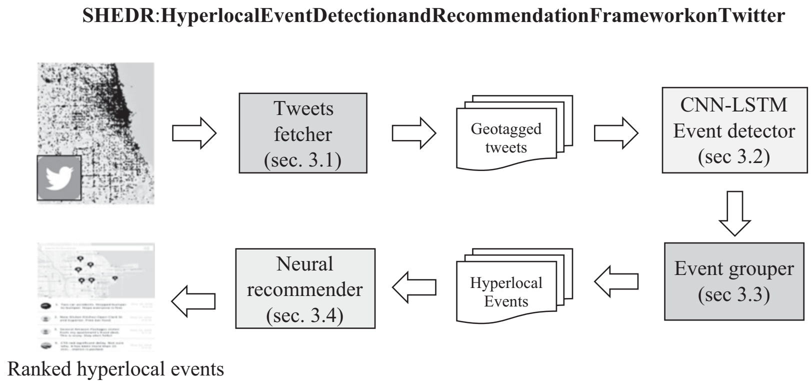
3. SHEDR: A Social Media–Based, End-to-End Neural Hyperlocal Event Detection and Recommendation Framework

In this section, we present SHEDR, an end-to-end deep learning–based hyperlocal event detection and recommendation framework. The overview of SHEDR is shown in Figure 1. In the following sections, we discuss each of the four components of the SHEDR framework, with a particular focus on the detector and the recommender.

3.1. Fetcher

The first component of the SHEDR framework is the fetcher. Given that the fetcher is not the focus of our technical contribution, we provide a succinct description of this component. Within the fetcher, we first tessellate a region into small and disjoint cells for retrieving geotagged tweets. There are several motivations behind this. First, a hyperlocal event, as its name implies, typically occurs at a specific small area and has limited spatial impact to other areas. Moreover, practically, by leveraging tessellation, we can aggregate geotagged tweets for each cell easily under Twitter's API restriction (900 tweets per user every 15

Figure 1. Workflow of SHEDR Framework



Note. The framework has four main components: (1) a fetcher to retrieve geotagged tweets; (2) an event detector to detect hyperlocal events from geotagged tweets; (3) a grouper to generate event clusters; and (4) a recommender to recommend hyperlocal events to end users.

minutes).[1] There are many tessellation methods such as grid, Voronoi, and square tessellation. Here, we use the state-of-the-art hierarchical triangular mesh (HTM) method, following the prior literature (Krumm and Horvitz 2015). The HTM method proposed by Kunszt et al. (2001) is a multilevel, recursive decomposition of the sphere that consists of equilateral triangles of nearly equal size (see Figure 2). It is widely used in spatial search, geographical indexing, and geocoding in commercial software systems (e.g., Xapiand[2]). Compared with other segmentation methods, HTM is more technically advanced and has several advantages. First, earth is a sphere, so HTM is naturally more fitting for segmenting a geographical region on the earth's surface than flat-based approaches such as grid segmentation. On the other hand, as Krumm and Horvitz (2015) also pointed out, density-based tessellation methods such as Voronoi diagrams use *k*-means clustering to form Voronoi regions that tend to be smaller for densely populated areas and larger otherwise. In contrast, HTM makes no implicit assumptions about how the extent of an event might vary with population or any other factor, and thus is less biased. In practice, we set the resolution of HTM at level 11, referring to the size of the mesh. It is calculated using the formula $(6{,}371{,}008.8 \times 2 \times PI/4)/(2^{level})$ where 6,371,008.8 is the diameter of earth. The higher levels correspond to higher resolution (i.e., smaller sizes) but also require more computation. We follow Krumm and Horvitz (2015) to use level 11 as the parameter, as it is shown to result in the best performance. Upon completion of the HTM tessellation, we use Twitter's API to retrieve geotagged tweets for each cell. It is notable that, whereas we use HTM to segment geographical regions, our key contribution, the SHEDR framework, can work with any tessellation methods, as these

Figure 2. Tweet Count distribution Around Chicago at 11 p.m.–12 a.m. (left) and 6–7 p.m. (right), July 15, 2018

Note. A number in an HTM cell refers to the value of tweet count at that time interval, and an empty grid cell means no tweets.

geographical segments are mainly used to help retrieve and aggregate geotagged tweets more effectively under Twitter's API restriction.

3.2. Detector

Next, to detect hyperlocal events, we formulate the following hyperlocal event detection problem: let Δt be a time interval, and let $S^p_{t,\,\Delta t+t}$ be a set of geotagged tweets posted during Δt in a region p. For each tweet $s \in S^p_{t,\,\Delta t+t}$, it is associated with a tuple $\langle tID, time, locationID, c\rangle$, where tID is the time stamp of the tweet, $locationID$ is the geographical location of the tweet in terms of latitude/longitude coordinates within p, and c is the text content of the tweet. We want to determine whether a hyperlocal event occurs in $S^p_{t,\,\Delta t+t}$ and extract and identify event-related tweets from $S^p_{t,\,\Delta t+t}$ based on their geographical, temporal, and semantic patterns.

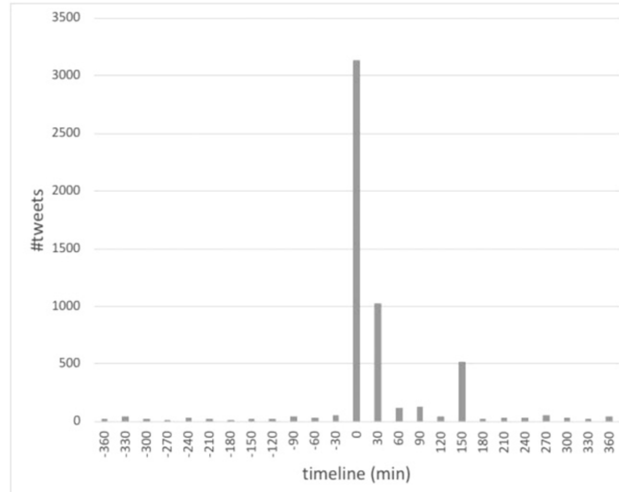
Our model is based on the anomaly detection on tweet volume. The prior frameworks that follow this approach include Eyewitness (Krumm and Horvitz 2015) and DeLLe (Wei et al. 2019). We build our model based on this approach, primarily because it is much more efficient than alternative content-based approaches (see discussion in the literature review). This is important, because many local residents often seek trendiness in hyperlocal events. To incorporate the content of the tweets, we later leverage the content data to filter out non-event-related tweets.

3.2.1. Our Model. One key observation that we build upon is that the occurrence of a hyperlocal event tends to generate an unusually higher number of tweets at or around the event location during a period of time. For example, a football game between the Seattle Seahawks and Chicago Bears that begins at 7 p.m. at Soldier Field in Chicago can suddenly draw increased attention and rapidly generate a large number of tweets with keywords such as "#bears," "#hawks," and "#chicago," posted in or around the field during the course of the game. By contrast, there is normally a small tweet volume around Soldier Field (see Figure 3).

Based on this observation, we first seek to predict the tweet volume for each region. If the actual volume is significantly higher than the prediction, then we consider that there is likely a spike that may correspond to a hyperlocal event. Formally, we need to identify whether $y_{p,t+\Delta t} - \widehat{y_{p,t+\Delta t}} > \delta$, where $y_{p,t+\Delta t}$ is the actual tweet volume posted during $[t, \Delta t + t]$ in a region p, $\widehat{y_{p,t+\Delta t}}$ is the predicted "normal" tweet volume during $[t, t + \Delta t]$ in p, and δ is the threshold.

The technical challenge of predicting tweet volume from a region at a given time period is largely caused by the complex spatial and temporal dependencies. Specifically, the time series of tweet volume can have very strong temporal dynamics. For example, there

Figure 3. Tweets Mentioning "#bears" Near Soldier Field (Home of the Chicago Bears) During a Game Day



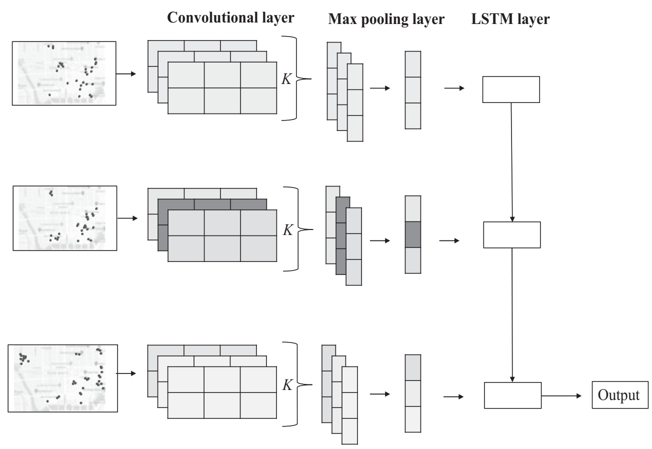
Note. A spike in tweet volume is observed.

could be periodic patterns, such as bursts near a coffee shop in the morning. Meanwhile, there could also be spikes about incidents such as car accidents that are totally nonstationary, making it difficult to predict. On the other hand, regions may have spatial correlations. For example, the area of bars and restaurants near a stadium may often see bursty tweets before and during a game and especially after the game is over.

Research on tweet volume prediction with spatiotemporal dynamics is still limited. The closest literature that we can identify is time-series prediction for traffic (Min and Wynter 2011). The traditional approach is to use autoregressive regression methods such as ARIMA (Moreira-Matias et al. 2013). Recent developments in deep learning have motivated a few recent studies to use deep learning methods to model spatial and temporal dynamics. Most recently, Wei et al. (2019) proposed a local event detection method on Twitter using convolutional LSTM. However, this method does not consider spatial and temporal dynamics simultaneously.

In this work, we propose a novel joint CNN-LSTM model for predicting the tweet volume from a region with the consideration of both spatial and temporal dynamics. The architecture of our model is presented in Figure 4. It is worth noting that, whereas CNN-LSTM architectures have recently been proposed and applied to several domains such as computer vision, natural language processing, and healthcare (e.g., Donahue et al. 2015, Wang et al. 2016), to the best of our knowledge, ours is the first work in the literature to develop a CNN-LSTM model for hyperlocal event detection space. More importantly, previous CNN-LSTM models rely on CNN models to extract global representations for the

Figure 4. Our Joint CNN-LSTM Model for Tweet Volume Prediction



input and subsequently feed them to LSTM models. For example, Donahue et al. (2015) used CNN to construct global feature vectors for the input images and LSTM to capture the spatial and temporal connection between these vectors for activity recognition. In our scenario, due to spatial dependencies, activities in neighboring regions are more likely to have a greater impact than distant regions on the target region. However, the global representation CNN or its variants, such as 3D CNN, are unable to model this spatial correlation properly. Therefore, we propose a localized CNN architecture to construct local presentations to characterize the spatial influence of nearby neighbors to the target region of interest. We next present details of this model.

3.2.2. Convolutional and Pooling Layer for Spatial Dependencies. In the CNN layer, due to spatial dependencies, we consider only spatially nearby regions at each time interval Δt. Inspired by prior works that capture spatial dynamics of an entire city's traffic as an image (Ma et al. 2017), for each time interval Δt, we treat a region p (i.e., an HTM cell) with its surrounding neighbors as one $S \times S$ image with one channel, in which the tweet volume from p and its neighbors during $[t, t+\Delta t]$ is retained. Note that p is the center of the image. The size of the image S corresponds to the spatial granularity. In practice, we set $S = 3$, covering roughly the eight closest neighbors of p. We also apply zero padding for regions at boundaries. Finally, we can construct a series of images corresponding to tweet volumes until time t as $[Y_{p,1}, \ldots Y_{p,t-2\Delta t} \cdots Y_{p,t-\Delta t}, Y_{p,t}]$. The localized CNN then uses this series as input, and each convolution layer is defined as

$$Y^k_{p,t} = ReLU(W^k_t \times Y^{k-1}_{p,\Delta t} + b^k_t),$$

where $Y^0_{p,t}$ is the image representing tweet volume at any given time t for the first convolutional layer, and so on and so forth. The parameters W^k_t and b^k_t are learned from data for the kth convolution layer, and they are shared across all regions to make the computation traceable. The notation $\times$ denotes the convolutional operation. We use ReLU, the rectifier function, defining $f(z) = max(0, z)$ as the activation function.

After K convolution layers, we use the max pooling layer to down-sample and consolidate the features learned in the previous layers. We also use a dropout layer (Srivastava et al. 2014) after the convolutional and max-pooling layers to address overfitting issues. After that, we use a flattened layer to generate a feature vector $\widehat{S_{p,t}}$ based on the output $Y^K_{p,t}$ for region p at time t. Finally, we use a fully connected layer to further consolidate and output the feature vector $S_{p,t} = ReLU(W^{fc}_t \widehat{S_{p,t}} + b^{fc}_t)$, where W^{fc}_t and b^{fc}_t are learnable parameters at time t.

3.2.3. LSTM Layer for Temporal Dependencies. To handle temporal dependencies among tweet volumes in a region over time, we propose to use long short-term memory (LSTM). Whereas the 3D CNN model may model spatial and temporal aspects of tweet volumes as a whole, it is important to note that such a solution may suffer data sparsity issues, as there may be no tweets posted during a specific time in a specific area. Instead, since we model spatiality and temporality separately in our joint CNN-LSTM model, this problem can be avoided. Specifically, LSTM maintains a memory cell c_t for time t, which is used to store information prior to t. The input gate i_t controls the new information accumulated at c_t at time t. The forget gate f_t determines whether to wipe the past information or store it on the memory cell at the previous time $t-1$. Finally, the output gate o_t controls which information should be used for the output of the memory cell. Here, we use the following equations to describe LSTM used in our model for temporal dependencies:

Gate

$$i_{p,t} = \sigma\Big(W_i g_{p,t} + U_i h_{p,t-\Delta t} + b_i\Big)$$
$$f_{p,t} = \sigma\Big(W_f g_{p,t} + U_f h_{p,t-\Delta t} + b_f\Big)$$
$$o_{p,t} = \sigma\Big(W_o g_{p,t} + U_o h_{p,t-\Delta t} + b_o\Big)$$

Transformation

$$\theta_{p,t} = tanh(W_g g_{p,t} + U_g h_{p,t-\Delta t} + b_g)$$

State update

$$c_{p,t} = f_{p,t} \odot c_{p,t-\Delta t} + i_{p,t} \odot \theta_{p,t}$$
$$h_{p,t} = o_{p,t} \odot tanh(c_{p,t})$$

where f, i, and o are gate vectors, σ denotes the sigmoid function, $\odot$ is the element-wise product, tanh is the hyperbolic tangent function, and W, U, b are parameters learned from training. Note that the LSTM part takes the output representation from the localized CNN $S_{p,t}$ as the input. The output of region p of LSTM after h time intervals is $h_{p,t}$.

3.2.4. Output Layer. This layer outputs the final result from our CNN-LSTM model for tweet volume prediction for region p during $[t, t+\Delta t]$. Since the value about tweet volumes is continuous, the prediction task requires a regression model. Therefore, instead of using a softmax classifier, we use a linear decoder in the final fully connected layer. We define it as

$$y_{p,t+\Delta t} = W_d h_{p,t} + b_d,$$

where $h_{p,t}$ is the vector learned from the LSTM layer that contains the influence of both temporal and spatial dependencies on region p at time t, y is the predicted tweet volume for region p in the next time interval $[t, t+\Delta t]$, and W_d and b_d denote the weight and bias associated with the linear decoder.

To train the model, given a training set $X = \{x_{p_1^t}, x_{p_1^{t+1}} \ldots x_{p_1^{t+m}}, x_{p_2^t}, \ldots x_{p_n^{t+m}}\}$, and their corresponding tweet volume $Y = \{y_{p_1^t}, y_{p_1^{t+1}} \ldots y_{p_1^{t+m}}, y_{p_2^t}, \ldots y_{p_n^{t+m}}\}$, we define the loss function

$$L(\theta) = \frac{1}{2mn} \sum_{i=1,j=1}^{m,n} \|h(x_{i,j}) - y_{i,j}\|^2,$$

where θ are all learnable parameters in our model. We use Adam (Kingma and Ba 2014) for optimization, and Tensorflow and Keras to implement our proposed model.

In practice, as we discussed before, sometimes the spatiotemporal patterns of tweet volumes may be periodic (e.g., unusual tweet volumes every morning near a coffee shop), which should not be mistakenly classified as a local event. To capture the periodic time-varying changes and novelty of the event, instead of measuring the difference between $y_{p,t+\Delta t}$ and $\widehat{y_{p,t+\Delta t}}$, we measure the normalized prediction difference as

$$\epsilon = \frac{y_{p,t+\Delta t} - \widehat{y_{p,t+\Delta t}}}{SD(h,d,w)},$$

where $SD(h,d,w)$ is the standard deviation of prediction differences on week w, hour h, and day d over region p during $[t, t+\Delta t]$. In practice, we use data from the prior eight weeks to measure this normalized prediction difference. We varied the threshold for ϵ from 0.5 to 5.0 and find that 3.0 achieves the best accuracy in detecting events.

3.3. Grouper

Our event detector returns geotagged tweets posted from the region of interest where spatiotemporal unusualness is detected. It is possible that these tweets may correspond to multiple events occurring in the same region around the same time. To separate these events after event detection, we propose an incremental clustering algorithm to cluster the event-associated tweets into event clusters effectively. One benefit of our clustering algorithm is that it does not require prior knowledge of the number of clusters. Instead, it assigns a tweet to a new cluster if other clusters are not similar to the tweet. Otherwise, this tweet is assigned to the cluster with the highest similarity. In order to measure the similarity, we consider several types of features of tweets and event clusters, including the following:

- temporal features: weekend or weekday, day of week;
- social features: count of mentions and replies, count of retweets, count of likes;
- tweet-centric features: topics of tweets and clusters using the topical model LDA (Blei et al. 2003), the tweet length, and 2-grams with TF-IDF scores.

In practice, we first construct these features for the tweet and the event cluster. Then, during the clustering process, we measure the distance between tweet t and cluster C_i based on these features as $sim(t, C_i) = \frac{1}{|C_i|}\sum_{t_j \in C_i} dist(t, t_j)$, where $dist(t, t_j)$ is the Euclidean distance between tweet t and t_j from cluster C_i.

Once the event clusters are created, we apply a classification model to each event cluster to determine whether a cluster indeed represents a local event. This step can potentially improve our event detection algorithm's accuracy beyond our joint CNN-LSTM model for detecting spatiotemporal anomalies. Specifically, we compute the aforementioned features for each event cluster. In addition, we measure the volume of tweets within each cluster as well as the prediction difference $\hat{\epsilon}$. We then train and learn a logistic regression classifier with these features to predict whether the cluster actually represents a hyperlocal event. As a result, we can filter out nonevent clusters and return true hyperlocal events for recommendation.

3.4. Recommender

To make the hyperlocal events more accessible, we need to rank these detected events to identify and recommend the top matches for a local resident. Formally, we define the problem of hyperlocal event recommendation as follows. Assume that there is a set of users $U = \{u_1, u_2, \ldots u_m\}$ and a set of hyperlocal events $E = \{e_1, e_2, \ldots e_n\}$. For each event, let y_k^i denote u_i's interest on event e_k. The value of y_k^i is either 1 (relevant) or 0 (nonrelevant) with the ordinal relationship (i.e., 0 < 1). The objective of the recommendation is to generate a relevance score y such that

$$y_j^i > y_k^i \Longleftrightarrow u_i \text{ prefers event } e_j \text{ to } e_k.$$

Inspired by the recent promising results of applying deep learning to recommender systems (Zhang et al. 2019), we propose a deep learning–based personalized pairwise ranking module that extends generalized matrix factorization to provide both linearity and nonlinearity for predicting the relevance score y. We next present the design of our model.

3.4.1. A Neural Model for Hyperlocal Events Recommendation.

The architecture of our model is shown in Figure 5. The input to our model is the user u_i, two hyperlocal events e_k and e_j, contextual features about these events, as well as the user preferences. The structure of the model is thus symmetric vertically with the user in the middle and two events on each side. The input layer consists of vectors v_i, v_j, and v_k about the identity of the user and events through a binary sparse one-hot encoding, where 1 at the i th position of v_i indicates u_i, and so on.

Next, above the input layer is a fully connected embedding layer. The goal is to project the sparse representation of the user and events to a much denser representation. Note that the embedding dimensions for user and events are the same. Formally, we have embeddings as

$$s_i = P^T v_i,$$
$$s_j = Q^T v_j,$$
$$s_k = L^T v_k,$$

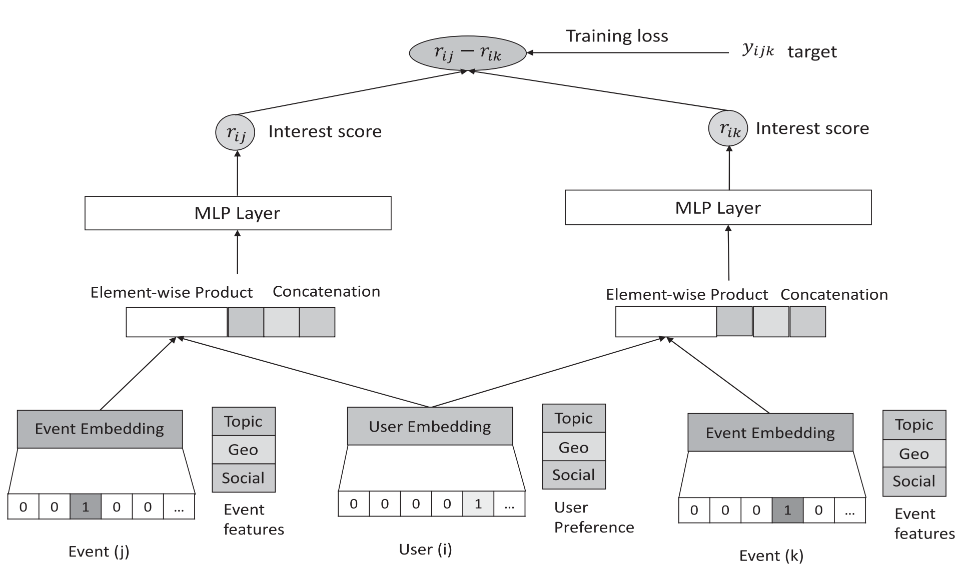
where $P \in R^{M\times K}$, $Q \in R^{N\times K}$, $L \in R^{N\times K}$ are embedding matrices for the user and two hyperlocal events. Note that, since we use one-hot encoding for users and hyperlocal events in the input layer, the obtained embedding vectors s_i, s_j, s_k essentially capture latent information about the user i and events j and k in a latent factor model. After that, analogous to the principles of matrix-factorization methods in recommender systems and, more recently, neural CF models (He et al. 2017), we perform element-wise product multiplication as

$$m_{ij} = s_i \odot s_j,$$
$$m_{ik} = s_i \odot s_k,$$

where $m_{ij} \in R^{M\times N}$ and $m_{ik} \in R^{M\times N}$ capture the two-way interaction of latent factors of user i and hyperlocal event j, and user i and hyperlocal event k, respectively, via linear combination.

As our goal is to provide personalized ranking of hyperlocal events to make them more accessible to the potential audience, in addition to the user and events, we also consider the contextual information to describe user preferences on events. There are several benefits of leveraging this auxiliary information. First of all, the model may face a sparsity issue, in the sense that a user may be interested in very few events. Therefore, it is challenging to recommend newly

Figure 5. Our Neural Model for Hyperlocal Event Recommendation



detected events to the user, since the previous implicit feedback from the user is limited. Second, as we discussed earlier, hyperlocal events are reported by and mainly consumed by local residents within a community. There may be many reasons why a resident is interested in a hyperlocal event; for example, the resident might be interested in the topic of the event, or the event just occurred near where the resident lives, or the resident has many friends who are interacting with others about the event on Twitter. Therefore, we further enhance our model by incorporating three types of contextual cues regarding topical, geographical, and social aspects of a hyperlocal event.

Formally, let T, G, S to capture user i's latent preference of topics of a hyperlocal event, geographic of the event, and social aspect of the event. Similar to the embedding layer, we have

$$q_i^T = T^T v_i,$$
$$q_i^G = G^T v_i,$$
$$q_i^S = S^T v_i,$$

where q_i^T, q_i^G, q_i^S capture the latent user preference for topics, geography, and social aspects for user i. Note that, similar to the embedding layer, the contextual preference hidden layer is also fully connected above user input and is to be learned. After that, we capture the interaction between user preference and the event's contextual features by performing element-wise multiplication to measure how well the event features and user preferences are aligned:

$$p_{ij}^{topics} = q_i^T \odot e_j^T,$$
$$p_{ij}^{geo} = q_i^G \odot e_j^G,$$
$$p_{ij}^{social} = q_i^S \odot e_j^S,$$

where e_j^T, e_j^G, e_j^S represent the topic, geography, and social features of the hyperlocal event j. Note that the setting is the same for the hyperlocal event k. (We discuss how to derive these contextual features in the next section.) Next, we concatenate the general preference of user i and event j, m_{ij} and the contextual preference of the event and user in the merge layer as

$$\widehat{m_{ij}} = \left[m_{ij}\ p_{ij}^{topics}\ \ p_{ij}^{geo}\ \ p_{ij}^{social}\right]^T.$$

Similarly, for user i and event k, we have

$$\widehat{m_{ik}} = \left[m_{ik}\ p_{ik}^{topics}\ \ p_{ik}^{geo}\ \ p_{ik}^{social}\right]^T.$$

We use standard multilayer perceptron (MLP) layers on the concatenated vector to learn the interaction between user and event latent features. As a result, our model can handle nonlinear, in addition to linear, interactions between the user and hyperlocal events. By contrast, the traditional way of matrix factorization–based recommendation considers only the linear relationship via the element-wise product on the user and events. Thus, our model is more flexible and has better expressiveness. More precisely, for event j and user i (the setting is similar for event k), we have

$$z_1 = \delta(\widehat{m_{ij}}),$$
$$\delta_2(z_1) = a(W_2^T z_1 + b_2),$$
$$\cdots$$
$$\delta_L(z_{L-1}) = a_L(W_L^T z_{L-1} + b_L).$$

Finally, we have, as the output of the predicted relevance score,

$$r_{ij} = \delta(h^T\ \delta_L(z_{L-1})),$$

where W is the weight matrix, b is the bias vector, and $\delta()$ is the activation function for the j th layer in the MLP. We use ReLU as the activation function, as it is popular in neural recommendation models.

To train the model, we need to minimize the pointwise loss between $r_{ij} - r_{ik}$ and its target value y_{ijk}. Specifically, our model takes user-event pairs as instances in learning and formulates the learning task as classification. As we mentioned earlier, let e_1^i and e_2^i be two events associated with u_i. The preference relationship $e_1^i > e_2^i$ means that e_1^i is more relevant than e_2^i with respect to u_i, which can be represented by $(r_{ij} - r_{ik}) > 0$. Otherwise, $(r_{ij} - r_{ik}) < 0$ if $x_1^i < x_2^i$. We transform the relevance into a set of preference instances with labels $y_i = +1$ (when $x_1^i > x_2^i$) and $y_i = -1$ (when $x_1^i < x_2^i$) as the product between the predicted difference $(r_{ij} - r_{ik})$ and ground truth. As a result, a larger value is acquired if their signs are aligned. Therefore, we have the following optimization formulation for training:

$$\max \sum_{i=1}^{l} \ln\left(\delta\left(y_i \cdot \left(x_1^i - x_2^i\right)\right)\right) - \lambda\|\theta\|^2.$$

In practice, we initialize the weight matrices with random values in [0, 1]. We use the Adam optimization algorithm (Kingma and Ba 2014) for gradient updates. Our model converges roughly 20 epochs given our training data.

3.4.2. Modeling Topical, Geographical, and Social Contextual Features. As we have discussed, our model incorporates three important contextual features. We next discuss these features and how we model them in detail.

Topical. The first important feature that we consider is the topical proximity between a target user's interest and a candidate event's description. We consider this feature following the engagement and endurability theories, which state that individuals are more willing to relive a previous good experience (O'Brien and Toms 2008). The application of this theory here indicates that a user may be interested in an event if its topics are similar to the topics for which the user has shown an interest in her or his tweets.

Two popular approaches that are used for learning a user's interest from Twitter data are the "content-centric" approach (e.g., Xu et al. 2011), wherein user interest is learned from content posted by the user, and the "following-centric" approach (e.g., Burgess et al. 2013), wherein user interest is learned from the user's following list. As the two approaches largely complement each other, we develop a hybrid approach that combines the advantages of both approaches to better measure the topical proximity. First, we collect past tweets posted by user u and event-related tweets of a local event e detected by our event detection algorithm. After standard text preprocessing, such as stemming and tokenization, we apply the topic model LDA to learn topic distributions Q_u and Q_e from u's past tweets and e's associated tweets, respectively. We set the number of topics K to 25, as that number achieves the best prediction performance in terms of the lowest perplexity score. As a result, we represent Q_u and Q_e as a vector of 25 topic distributions. Finally, we measure topical proximity between Q_e and Q_u by applying the symmetric Jensen–Shannon (JS) divergence: $Sim_{Content}(Q_e, Q_u) = \frac{1}{2}D_{KL}(Q_u||M) + \frac{1}{2}D_{KL}(Q_e||M)$, where $M = \frac{1}{2}(Q_u + Q_e)$ and $D_{KL}(Q_u||M) = -\sum_{1\le i\le K} Q_u(i)\frac{M(i)}{Q_u(i)}$ is the standard KL-diverge function. Intuitively, higher JS-divergence means that the user's topical interests are aligned with the event's topics.

In addition, we also develop a following-centric method for measuring the topical proximity. First, for each user that u follows on Twitter, we obtain her or his Twitter profile description. After combining these descriptions as a single text document, we learn a topic distribution Q_f from this document again using LDA, where we set the number of topics K to 25. We again calculate the topical proximity between Q_f and the topic distribution of event Q_e using JS-divergence, $Sim_{following}(Q_f, Q_e) = \frac{1}{2}D_{KL}(Q_f||M) + \frac{1}{2}D_{KL}(Q_e||M)$, where $M = \frac{1}{2}(Q_f + Q_e)$. This time, higher JS-divergence indicates that the user's topical interests—as reflected in the people whom the user is following—are closer to the event's topics.

Finally, we combine the content-centric and following-centric measures as a weighted combination to represent final topical proximity between user and event: $Sim_{topical}(e, u) = \alpha Sim_{Content}(T_e, T_u) + (1-\alpha) Sim_{following}(T_e, T_u)$, where α is the weight assigned to each measure to reflect its importance, and we set $\alpha = 0.38$ in practice.

Geographical. It is generally known that an individual's offline geographical location can affect her or his online behavior such as tie formation, interaction with other users, and information diffusion (Takhteyev et al. 2012). Based on prior findings and following the underlying principle in most location-aware recommender systems (Scellato et al. 2011), we assume that, ceteris paribus, a user would be more interested in that event as the user is closer to the event's location.

The location of the candidate event is already known when we detect the event using HTMs. Thus, we need to infer the user's location to measure the location proximity. Whereas inferring a user's location from locations of their friends or the content of their tweets is possible according to recent studies (Cheng et al. 2010), such inference usually comes at the resolution of a city or a state. This will cause a mismatch in measuring the location proximity, as our detected events are at a neighborhood level.

To address this challenge, we proactively ask the users to enter the top three neighborhoods that they most care about/are most interested in when they first use our recommender system. With user location as available information, let L_u be the set of centering latitude-longitude coordinates of places that the user prefers. Then, to measure the geographical proximity between user-preferred location L_u and the event's location L_e, we define the following formula:

$$Sim_{location}(u, e) = \frac{1}{|L_u|}\sum_{l_i \in L_u} K_{\mathbf{H}}(l_i, L_e),$$

where $K_{\mathbf{H}}(l_i, L_e) = \sigma^2 \exp(-||l_i - L_e||^2/2\sigma^2)$ is the Gaussian kernel.

Social. Our last cue is the social proximity between the user and the individuals who are already involved in the event on Twitter. It is known that a user's social network can significantly influence the user's behavior (Hong et al. 2018). In the area of event studies, Gil de Zúñiga and Valenzuela (2011) showed that there exists a positive relationship between individuals' online social network and their offline engagement. Their study further showed that social ties among individuals, in particular, the weak ties, are the key predictors of their involvement in civic issues. Social proximity also plays an important role in many recommendation tasks (Ye et al. 2012). Therefore, following the literature, we expect that the likelihood of the user u interested in an event e depends on the number of individuals who have weak ties with u who are already engaged with e on Twitter.

We define a *weak tie* on Twitter as a unidirectional link between two Twitter users. In other words, following a user or being followed by a user on Twitter can be considered as having a weak tie with that user. Based on this, we define the social proximity between user u and event e as

$$Sim_{social}(u, e) = \frac{|G_u \cap P_e|}{|G_u \cup P_e|},$$

where G_u is the group of individuals whom u is following on Twitter and P_e are the individuals who have already engaged with the event prior to u. This measure basically examines social proximity between u and e in terms of the overlaps between the weak ties of u and individuals involved in the event.

4. Performance Evaluation

In this section, we evaluate the performance of our proposed SHEDR framework. In the following subsections, we first present details of data collection. We then discuss the model settings, baselines, and evaluation metrics. Next, we report the performance of our event recommender with baseline models. In addition, we also examine the effect of contextual features on our model's performance. Finally, we investigate how our models can perform and converge with respect to different amounts of training data.

4.1. Data Collection

Our evaluation is based on two large sets of geotagged tweets collected from Seattle and Chicago. The total number of tweets is roughly 900,000 in the Chicago area and 450,000 in the Seattle area, in the period from June 23 to August 23, 2018. For the hyperlocal event detection task, since we need to predict tweet volume based on previous volumes, we split the data set based on the time stamp by using tweets posted in the first 45 days as training and validation (at a ratio of 0.2), for training our CNN-LSTM tweet volume prediction model, and using tweets posted in the last 15 days for testing. The same data set also applies to our grouper module. To construct the ground truth for hyperlocal events, we sampled 1,000 event clusters randomly from HTMs of each city using our algorithm. We then hired 108 annotators from Amazon Mechanical Turk, who are required to be proficient in the English language to obtain the ground truth for these events. Specifically, each detected hyperlocal event was presented with a sample of its associated tweets to its annotators. We also show a map displaying the location of the event along with these tweets. For each candidate event, we ask each annotator to determine if these tweets really are about a local event. The annotator needs to choose either yes, no, or unsure. Each event was annotated by three different annotators (\$0.02 per event, up to 200 events per annotator). Finally, 72.4% of events were determined to be hyperlocal events, 24.3% were determined to be nonlocal events, and the rest were "unsure," and so we ignored them in our evaluation. On the other hand, for the hyperlocal event detection task, we constructed the training and test data sets based on those events. We recruited 100 residents from Seattle and Chicago, respectively, from Amazon Mechanical Turk and asked them to label the detected events as interested (1) or not interested (0) individually. We paid annotators \$0.1 per event for up to 200 events per annotator. As a requirement for our recruitment, every participant should be an active Twitter user. Thus, we also obtained these individuals' Twitter profiles, as well as their interested neighborhoods in Seattle and Chicago. We later used this information for measuring topical, geographical, and social proximities. Finally, based on these labeled events, we can construct over 75,000 tuples (i, j, k) for training and testing our recommendation algorithm, where i, j, k corresponds to the user index and indexes for two hyperlocal events; in our setting, user i prefers j over k.

4.2. Model Setup

For the event detection detector, we set $K = 3$, $\tau = 3 \times 3$ for the filter size; in total, 20 filters were used. We use batch normalization for the localized CNN, and the batch size is 64. We set the dropout rate to 0.5. We set the sequence length to 4 in LSTM. For the event recommender, we set the user and event latent factor dimensions to 100. For the MLP, we use three hidden layers with layer dimensions 300, 200, and 100, respectively. We use mini-batch gradient descent with batch size as 512. We also tune the regularization parameters and set $\lambda = 5$. For both models, we apply early stopping to terminate training if there is no decrease on validation loss for three epochs, to avoid overfitting.

4.3. Baseline Approaches

We use the following baselines to evaluate the performance of our hyperlocal event detector:

- *EvenTweet* (Abdelhaq et al. 2013), which detects keywords that have bursty temporal and spatial patterns. The detected keywords then are grouped to find hyperlocal events.
- *Eyewitness* (Krumm and Horvitz 2015), which finds unusual spikes in tweet volume in local areas via a regression model and classifies them as hyperlocal events.
- *TF-IDF-based approach* (Naaman et al. 2011), which clusters tweets based on their textual content represented as TF-IDF vectors and characterizes the clusters as events or nonevents.
- *Wavelet-based approach* (Weng and Lee 2011), which explores a term's burstiness using the standard wavelet transform and calculates the pairwise correlation between bursty terms. Finally, it uses coherently connected bursty terms to signal an event's occurrence.
- *DeLLe* (Wei et al. 2019), which is a deep learning–based approach using LSTM.

Next, to evaluate our hyperlocal event recommender, we compare it to the following competing recommendation approaches commonly used in the literature:

- *Chronological*: We rank events based on their chronological order directly, which is the default way of tweet ranking on Twitter.

- *Retweet*: We rank events based on the number of times that they are retweeted/shared. This approach mainly focuses on the popularity of an event and ignores personalization.
- *Profiling*: We rank events based on the content similarity between event-related tweets (for the detected events data set) and the user's profile. This strategy is mainly used in modern content-based recommender systems (Pazzani and Billsus 2007).
- *Matrix factorization (MF)*: This is a popular method used in many collaborative filtering-based recommender systems. Here, we implement the state-of-the-art approached called *Sorec* (Ma et al. 2008).
- *Neural collaborative filtering (NCF)*: MLP-based neural recommendation model from (He et al. 2017). We set it with three hidden layers, with the size of the dimensions the same as ours.

4.4. Performance Metrics

To examine the performance of hyperlocal event detection, we use standard metrics such as precision, recall, and F-1 score measures. Precision measures the percentage of detected hyperlocal events that are indeed hyperlocal based on our labeled event corpus. Recall measures the fraction of actual hyperlocal events detected out of all hyperlocal local events. It is important to note that measuring recall is very difficult. This is because, in practice, we do not know how many actual hyperlocal events are out there in the real world, as there is no repository of ground truth of hyperlocal events. In fact, many hyperlocal events may go unreported (this actually reflects the value of tools like SHEDR that help local residents discover happenings in their community). Therefore, we follow the same practice of previous event detection studies (e.g., Krumm and Horvitz 2015, Wei et al. 2019) and have our "recall" and F-1 score calculated based on a pseudo ground truth of hyperlocal events constructed by combining a set of distinct true positive hyperlocal events reported by different baselines.

On the other hand, for evaluating the event recommender, we adopt popular metrics in the recommender system literature for evaluation, namely, precision at position k, recall at position k, and F-1 at position k. Basically, these measures are modifications of the standard precision, recall, and F-1 measure but reflect the importance of ranking in the recommendation results. Formally, we have that

$$P@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|GroundTruth(u_i) \cap List^k(u_i)|}{|k|},$$

$$R@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|GroundTruth(u_i) \cap List^k(u_i)|}{|GroundTruth(u_i)|},$$

$$F1@k = \frac{2 \cdot P@k + R@k}{P@k + R@k},$$

where $GroundTruth(u_i)$ is the ground truth of what user u_i prefers in the test data and $List^k(u_i)$ is the top k recommended hyperlocal event list for u_i.

Table 1. Performance Comparison of Event Detection on the Seattle Data Set

Method	Precision (gain)	Recall (gain)	F-1 score (gain)
EvenTweet	42.2 (112.3%)	40.2 (82.8%)	41.2 (96.1%)
Eyewitness	62.1 (44.3%)	61.2 (20.1%)	61.6 (31.2%)
DeLLe	72.3 (23.9%)	60.1 (22.3%)	65.6 (23.2%)
TF-IDF approach	51.4 (74.3%)	45.2 (62.6%)	48.1 (68.0%)
Wavelet approach	49.3 (81.7%)	43.3 (69.7%)	46.1 (75.3%)
Our method	89.6	73.5	80.8

4.5. Performance of Hyperlocal Event Detector

We begin by evaluating the performance of the hyperlocal event detector in Tables 1 and 2. We first observe that our model significantly improves EvenTweet, Eyewitness, TF-IDF, and Wavelet approaches in all the metrics. The performance of DeLLe is better than these baselines, but our method still outperforms it with an improvement of 23.9% on precision and 22.3% on recall for the Seattle data set and 27.6% on precision and 13.8% on recall for the Chicago data set, demonstrating the effectiveness of our joint CNN-LSTM model for detecting tweet volume unusualness with the characterizations of both spatial and temporal dependencies.

One key parameter in our model is the threshold ϵ, which controls the sensitivity of our unusualness detection. To evaluate how this parameter affects our model performance, we show the precision, recall, and F-1 measures when varying ϵ. Tables 3 and 4 show the results on two data sets. In general, the precision increases as ϵ increases. This is because a larger ϵ means a greater difference between the actual tweet volume and the predicted tweet volume, indicating that there were many more tweets than usual, which leads to a likely hyperlocal event. Accordingly, recall drops as ϵ increases. When $\epsilon = 3$, we find the highest F-1 score, which we used for event detection.

4.6. Performance of Hyperlocal Event Recommender

The performance comparison of the hyperlocal event recommendation results of P@k and R@k for the two

Table 2. Performance Comparison of Event Detection on the Chicago Data Set

Method	Precision	Recall	F-1 score
EventTweet	39.3 (118.3%)	39.1 (82.9%)	39.2 (99.0%)
Eyewitness	59.7 (43.7%)	60.3 (18.6%)	60.0 (30.0%)
DeLLe	67.2 (27.7%)	62.8 (13.9%)	64.9 (20.2%)
TF-IDF approach	52.3 (64.1%)	51.1 (39.9%)	51.7 (50.9%)
Wavelet approach	54.2 (58.3%)	52.7 (35.7%)	53.4 (46.1%)
Our method	85.8	71.5	78.0

Table 3. Sensitivity of Model Parameter on the Seattle Data Set

ϵ	Precision	Recall	F-1 score
0.5	68.2	80.2	73.7
1	75.4	77.2	76.3
1.5	79.6	76.5	78.0
2	79.7	75.5	77.5
2.5	85.1	74.2	79.3
3	89.6	73.5	80.8
3.5	90.2	70.2	79.0
4	90.7	54.2	67.9
4.5	91.2	40.2	55.8
5	92.8	34.2	50.0

data sets is shown in Tables 5–8. Overall, the simplest, chronological strategy approach achieves the worst performance. Also, recommendation based on retweet approach performs poorly. On the other hand, the content-based profiling method and the collaborative filtering method are slightly better in terms of precision at k and recall at k measure. The best baseline method is the state-of-the-art deep learning–based NCF approach which achieves an average of 72% precision and 70% recall on two data sets. In comparison, our approach models not just content but also location and social factors in a neural pairwise ranking model that achieves over an average of 15.2% and 13.1% improvements compared with NCF in terms of precision and recall at k. A similar pattern also holds on the Chicago data set, where our approach again achieves the best precision and recall at k, or 17.4% and 18.5% improvements, compared with NCF.

On the other hand, because our model comprises three major contextual features—topical proximity, social proximity, and geographical proximity—it is useful to examine the effectiveness of each feature and the robustness of our approach. Therefore, we perform several experiments by focusing on the performance of just one feature at a time and combine them later. We report the results in Tables 9–10 on the Seattle data set. The results on the Chicago data set are similar. It is clear that topical proximity is the most

Table 4. Sensitivity of Model Parameter on the Chicago Data Set

ϵ	Precision	Recall	F-1 score
0.5	66.2	83.4	73.8
1	73.4	80.5	76.8
1.5	76.8	77.5	77.1
2	79.2	75.3	77.2
2.5	81.1	73.3	77.0
3	85.8	71.5	78.0
3.5	90.2	66.2	76.4
4	91.2	54.8	68.5
4.5	91.8	42.3	57.9
5	93.1	40.6	56.5

helpful single feature in terms of achieving higher precision and recall than social proximity and geographical proximity, which is expected, as residents often view the event content before deciding to engage with it or not. On the other hand, social proximity plays a more important role in recall and geographical proximity, while the latter is more important to the performance of prediction. Nevertheless, we achieve the best recommendation performance when we combine all the features together, resulting in an average improvement of 20% on precision and an average improvement of 15.6% on recall.

Finally, Figure 6 shows the convergence of our model in terms of F1@k. Our model converges at around 20 rounds, or approximately 10 minutes, indicating its robustness and efficiency.

5. General Discussion

Social media have emerged as significant and important sources of information for individuals and organizations. Compared with traditional media, social media have many advantages in terms of creating and disseminating content (Hong et al. 2018) and have become the de facto platforms to satisfy individuals' needs for information and social interactions. However, social media also tend to have the issue of noisiness of information and overload for user cognitive capacity. We seek to design and implement novel methods to identify, extract, summarize, and recommend events for local residents in the noisy and overloaded web environment, which has increasingly become one of the most important research objectives on the intersection of information systems and computer science (Gregor and Hevner 2013). Designing algorithms and systems to provide actionable insights to individuals about hyperlocal events is important for individuals' life quality and society's public discourse and, therefore, is highly relevant to the emerging research on using analytics to address societal issues. In this study, we design and evaluate SHEDR, a novel end-to-end deep learning–based hyperlocal event detection and recommendation framework to help keep residents aware of and engaged with relevant and important community events.

Our approach builds on and significantly extends prior work in the intersecting information systems and social computing literatures (Adomavicius and Tuzhilin 2011, Krumm and Horvitz 2015, Tim et al. 2017). Specifically, we first develop a novel event detection model based on joint CNN-LSTM that is capable of effectively characterizing spatiotemporal dependencies that capture unusual spikes of tweets in a region of interest. Specifically, in previous CNN-LSTM models for computer vision, natural language processing, and healthcare (e.g., Donahue et al. 2015, Wang et al.

Table 5. Recommendation Performance (%) on Precision on the Seattle Data Set

	P@1	P@3	P@5	P@10
Chronological	28.5 (210.2%)	26.7 (213.1%)	17.8 (360.7%)	12.8 (518.8%)
Retweet	28.9 (205.9%)	31.2 (167.9%)	31.3 (162.0%)	35.1 (125.6%)
Profiling	51.2 (72.7%)	48.1 (73.8%)	45.6 (79.8%)	45.5 (74.1%)
MF	71.1 (24.3%)	68.2 (22.6%)	64.3 (27.5%)	61.2 (29.4%)
NCF	75.4 (17.2%)	73.1 (14.4%)	71.8 (14.2%)	68.9 (14.9%)
Our method	88.4	83.6	82.0	79.2

Table 6. Recommendation Performance (%) on Recall on the Seattle Data Set

	R@1	R@3	R@5	R@10
Chronological	26.2 (222.1%)	25.4 (222.4%)	14.4 (432.6%)	13.4 (438.8%)
Retweet	27.5 (206.9%)	26.6 (207.9%)	24.3 (215.6%)	23.6 (205.9%)
Profiling	55.5 (52.1%)	56.3 (45.5%)	55.8 (37.5%)	53.8 (34.2%)
MF	67.5 (25.0%)	63.2 (29.6%)	58.2 (31.8%)	53.9 (34.0%)
NCF	77.3 (9.2%)	71.3 (14.9%)	67.6 (13.5%)	62.4 (15.7%)
Our method	84.4	81.9	76.7	72.2

Table 7. Recommendation Performance (%) on Precision on Chicago Data set

	P@1	P@3	P@5	P@10
Chronological	18.3 (387.4%)	17.7 (396.6%)	18.8 (337.2%)	12.7 (518.1%)
Retweet	32.1 (177.9%)	34.0 (158.5%)	33.8 (143.2%)	35.4 (121.8%)
Profiling	52.2 (70.9%)	48.1 (82.7%)	46.1 (78.3%)	45.3 (73.3%)
MF	71.5 (24.8%)	68.3 (28.7%)	63.3 (29.9%)	61.4 (27.9%)
NCF	76.5 (16.6%)	74.4 (18.1%)	71.1 (15.6%)	66.7 (17.7%)
Our method	89.2	87.9	82.2	78.5

Table 8. Recommendation Performance (%) on Recall on the Chicago Data Set

	R@1	R@3	R@5	R@10
Chronological	16.2 (439.5%)	15.4 (442.2%)	14.4 (460.4%)	13.4 (479.1%)
Retweet	27.5 (217.8%)	26.6 (213.9%)	24.3 (232.1%)	23.6 (228.8%)
Profiling	45.5 (92.1%)	46.3 (80.3%)	45.8 (76.2%)	43.8 (77.2%)
MF	67.5 (29.5%)	63.2 (32.1%)	58.2 (38.7%)	53.9 (44.0%)
NCF	72.5 (20.6%)	70.3 (18.8%)	68.6 (17.6%)	66.4 (16.9%)
Our method	87.4	83.5	80.7	77.6

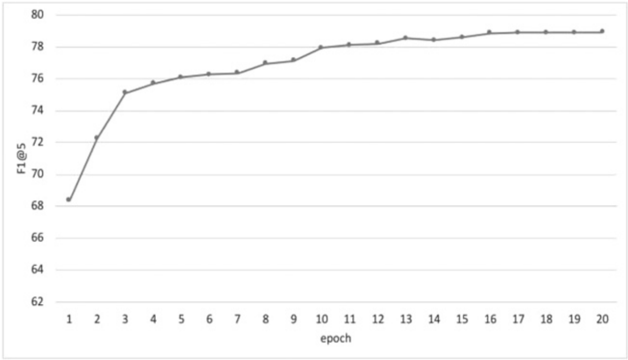
Table 9. Effect of Contextual Features (%) on Precision on the Seattle Data Set

	Topical	Geographical	Social	Topical + Geographical + Social
P@1	77.8 (13.6%)	74.8 (18.2%)	69.2 (27.7%)	88.4
P@3	76.5 (9.3%)	73.5 (13.7%)	63.8 (31.0%)	83.6
P@5	74.2 (10.5%)	68.4 (19.9%)	63.4 (29.3%)	82.0
P@10	69.1 (14.6%)	66.2 (19.6%)	59.3 (33.6%)	79.2

Table 10. Effect of Contextual Features (%) on Recall on the Seattle Data Set

	Topical	Geographical	Social	Topical + Geographical + Social
R@1	74.7 (13.0%)	69.1 (22.1%)	72.2 (16.9%)	84.4
R@3	72.7 (12.7%)	67.5 (21.3%)	69.8 (17.3%)	81.9
R@5	69.2 (10.8%)	64.4 (19.1%)	66.4 (15.5%)	76.7
R@10	67.5 (7.0%)	59.5 (21.3%)	65.3 (10.6%)	72.2

Figure 6. Convergence of Our Event Recommendation Model



2016), CNN is used to extract global representations for the input and subsequently feed them to LSTM models. In scenarios similar to ours, spatial dependencies need to be addressed, as activities in neighboring regions are more likely to have a greater impact on the target region than activities in distant regions, according to the first law of geography. Prior models with the global representation CNN are unable to properly capture this spatial correlation. By contrast, our CNN-LSTM model uses a new localized CNN architecture that constructs local feature representations only for spatially nearby regions. In this way, spatial dependencies are better retained. For evaluation, our novel model significantly outperforms other state-of-the-art existing approaches. Next, we extend prior research on context-aware recommendation systems (Adomavicius and Tuzhilin 2011, Panniello et al. 2016) by building a deep learning–based hyperlocal event recommender based on pairwise ranking, exploiting several proximity-based contextual signals, such as topical proximity between the user's topic preference and the event' description, social proximity based on the user's and the event participants' profiles, and geographical proximity based on the user's location and the event's location. The evaluation results show that contextual signals can help significantly improve hyperlocal event recommendation performance by a margin of over 15% over baseline methods in terms of both precision and recall.

Our research can be applied to practical settings. The methods proposed can be used for various event detection and recommendation tasks to promote awareness, strengthen bonds among community members, and cultivate a culture of civic engagement (Galliers 2011). In the context of citizen journalism, an uptrend phenomenon outside mainstream news production circles, resident reporters mainly use journalistic practices that are similar to those used by professional journalists, but they rely more on alternative information sources, such as social media, than traditional or mainstream journalists (Radsch 2013). Our research demonstrates to practitioners the effectiveness of novel automated deep learning approaches in harnessing social media's digital capabilities for distilling and recommending newsworthy information in the noisy social media context.

Given the paramount importance of hyperlocal community and civic engagement in society, we believe that our research provides actionable techniques and state-of-the-art frameworks that can help enhance civic engagement. It is our hope that this is a mere first step toward more research that focuses on high-performance event detection and context-aware recommendation systems, which have the potential to enhance civic engagement for a healthy and prosperous society.

6. Data Set

In accordance with the *Journal on Computing*'s data policy, we are sharing the data set used in the paper at https://henryyhu.github.io/jocdata2.zip.

Endnotes

[1] See https://developer.twitter.com/en/docs/twitter-api/v1/rate-limits.

[2] See https://kronuz.io/Xapiand/tutorials/spatial-search/#searching.

References

Abdelhaq H, Sengstock C, Gertz M (2013) EvenTweet: Online localized event detection from Twitter. *Proc. VLDB Endowment* 6(12): 1326–1329.

Adomavicius G, Tuzhilin A (2011) Context-aware recommender systems. Ricci F, Rokach L, Shapira B, Kantor PB, eds. *Recommender Systems Handbook* (Springer, Boston), 217–253.

Atefeh F, Khreich W (2015) A survey of techniques for event detection in Twitter. *Comput. Intelligence* 31(1):132–164.

Becker H, Iter D, Naaman M, Gravano L (2012) Identifying content for planned events across social media sites. *Proc. 5th ACM Internat. Conf. Web Search Data Mining* (ACM, New York), 533–542.

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. *J. Machine Learning Res.* 3:993–1022.

Bowman S, Willis C (2003) We media: How audiences are shaping the future of news. Report, American Press Institute, Reston, VA.

Bruns A, Wilson JA, Saunders BJ (2008) Building spaces for hyperlocal citizen journalism. *Proc.* Internet Res. 9.0: Rethinking Community, *Rethinking Place*, *Copenhagen*, 15–18.

Burgess M, Mazzia A, Adar E, Cafarella MJ (2013) Leveraging noisy lists for social feed ranking. *Proc. 7th Internat. AAAI Conf. Weblogs Social Media* (AAAI Press, Palo Alto, CA), 31–40.

Carroll JM, Horning M, Hoffman B, Ganoe C, Robinson H, Rosson MB (2011) Visions, participation and engagement in new community information infrastructure. *J. Community Informatics* 7(3).

Cheng Z, Caverlee J, Lee K (2010) You are where you Tweet: A content-based approach to geo-locating twitter users. *Proc. ACM Internat. Conf. Inform. Knowledge Management* (ACM, New York), 759–768.

D'heer E, Paulussen S (2013) The use of citizen journalism for hyperlocal news production. *Rech. Comm.* 39:151–164.

Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. *Proc. IEEE Conf. Comput. Vision Pattern Recognition* (IEEE, Piscataway, NJ), 2625–2634.

Farnham SD, Lahav M, Monroy-Hernandez A, Spiro E (2015) Neighborhood community well-being and social media. *Environments* 40:49.

Galliers RD (2011) In celebration of diversity in information systems research. *J. Inform. Tech.* 26(4):299–301.

Gil de Zúñiga H, Valenzuela S (2011) The mediating path to a stronger citizenship: Online and offline networks, weak ties, and civic engagement. *Comm. Res.* 38(3):397–421.

Gregor S, Hevner AR (2013) Positioning and presenting design science research for maximum impact. *MIS Quart.* 37(2):337–356.

Han K, Shih PC, Carroll JM (2014) Local news chatter: Augmenting community news by aggregating hyperlocal microblog content in a tag cloud. *Internat. J. Human Comput. Interaction* 30(12):1003–1014.

He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. *Proc. 26th Internat. Conf. World Wide Web, Perth, Australia*, April 3–7, 173-182.

Hong Y, Hu Y, Burtch G (2018) Embeddedness, pro-sociality, and social influence: Evidence from online crowdfunding. *MIS Quart.* 42(4):1211–1224.

Hu Y, Farnham SD, Monroy-Hernández A (2013) Whoo.ly: Facilitating information seeking for hyperlocal communities using social media. *Proc. SIGCHI Conf. Human Factors Comput. Systems* (ACM, New York), 3481–3490.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. Preprint, submitted December 22, https://arxiv.org/abs/1412.6980.

Krumm J, Horvitz E (2015) Eyewitness: Identifying local events via space-time signals in twitter feeds. *Proc. 23rd SIGSPATIAL Internat. Conf. Adv. Geographic Inform. Systems* (ACM, New York), 20.

Kunszt PZ, Szalay AS, Thakar AR (2001) The hierarchical triangular mesh. Banday AJ, Zaroubi S, Bartelmann M, eds. *Mining the Sky* (Springer, Berlin), 631–637.

Lee R, Sumiya K (2010) Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection. *Proc. ACM SIGSPATIAL Internat. Workshop Location Based Soc. Networks* (ACM, New York), 1–10.

Liu X, He Q, Tian Y, Lee WC, McPherson J, Han J (2012) Event-based social networks: Linking the online and offline social worlds. *Proc. ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (ACM, New York), 1032–1040.

López C, Farzan R (2015) From community networks to hyper-local social media. *Proc. ACM Conf. Companion Comput. Supported Cooperative Work Soc. Comput.* (ACM, New York), 239–242.

Ma H, Yang H, Lyu MR, King I (2008) SoRec: Social recommendation using probabilistic matrix factorization. *Proc. ACM Conf. Inform. Knowledge Management* (ACM., New York), 931–940.

Ma X, Dai Z, He Z, Ma J, Wang Y, Wang Y (2017) Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17(4):818.

Masden CA, Grevet C, Grinter RE, Gilbert E, Edwards WK (2014) Tensions in scaling-up community social media: a multi-neighborhood study of nextdoor. *Proc. ACM Conf. Human Factors Comput. Systems* (ACM, New York), 3239–3248.

Metzgar ET, Kurpius DD, Rowley KM (2011) Defining hyperlocal media: Proposing a framework for discussion. *New Media Soc.* 13(5):772–787.

Min W, Wynter L (2011) Real-time road traffic prediction with spatio-temporal correlations. *Transportation Res. Part C Emerging Tech.* 19(4):606–616.

Moreira-Matias L, Gama J, Ferreira M, Mendes-Moreira J, Damas L (2013) Predicting taxi–passenger demand using streaming data. *IEEE Trans. Intelligent Transportation Systems* 14(3):1393–1402.

Naaman M, Becker H, Gravano L (2011) Hip and trendy: Characterizing emerging trends on Twitter. *J. Amer. Soc. Inform. Sci. Tech.* 62(5):902–918.

O'Brien HL, Toms EG (2008) What is user engagement? A conceptual framework for defining user engagement with technology. *J. Amer. Soc. Inform. Sci. Tech.* 59(6):938–955.

Panniello U, Gorgoglione M, Tuzhilin A (2016) In CARSs we trust: How context-aware recommendations affect customers' trust and other business performance measures of recommender systems. *Inform. Systems Res.* 27(1):182–196.

Pazzani MJ, Billsus D 2007. Content-based recommendation systems. Brusilovsky P, Kobsa A, Nejdl W, eds. *The Adaptive Web: Methods and Strategies of Web Personalization* (Springer, Berlin), 325–341.

Putnam RD (2000) Bowling alone: America's declining social capital. Crothers L, Lockhart C, eds. *Culture and Politics* (Palgrave Macmillan, New York), 223–234.

Radsch CC (2013) The revolutions will be blogged: Cyberactivism and the 4th Estate in Egypt. Unpublished doctoral dissertation, American University, Washington, DC.

Scellato S, Noulas A, Lambiotte R, Mascolo C (2011) Socio-spatial properties of online location-based social networks. *Proc. 5th Internat. AAAI Conf. Weblogs Social Media* (AAAI Press, Palo Alto, CA), 329–336.

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *J. Machine Learning Res.* 15(1):1929–1958.

Stolle D, Hooghe M (2005) Inaccurate, exceptional, one-sided or irrelevant? The debate about the alleged decline of social capital and civic engagement in Western societies. *British J. Political Sci.* 35(1):149–167.

Takhteyev Y, Gruzd A, Wellman B (2012) Geography of Twitter networks. *Soc. Networks* 34(1):73–81.

Tezon AL (2003) Cheerleaders, watchdogs and community builders: How rural weekly newspaper publishers in the Midwest view their roles. Unpublished doctoral dissertation, University of Memphis, Memphis, TN.

Tim Y, Pan SL, Bahri S, Fauzi A (2017) Digitally enabled crime-fighting communities: Harnessing the boundary spanning competence of social media for civic engagement. *Inform. Management* 54(2):177–188.

Tobler WR (1970) A computer movie simulating urban growth in the Detroit region. *Econom. Geography* 46:234–240.

Wang J, Yu LC, Lai KR, Zhang X (2016) Dimensional sentiment analysis using a regional CNN-LSTM model. *Proc. 54th Annual Meeting Assoc. Comput. Linguistics*, vol. 2 (Association for Computational Linguistics, Stroudsburg, PA), 225–230).

Wei H, Zhou H, Sankaranarayanan J, Sengupta S, Samet H (2019) DeLLe: Detecting latest local events from geotagged tweets. *Proc. 3rd ACM SIGSPATIAL Internat. Workshop Analytics Local Events News* (ACM, New York), 1–10).

Weng J, Lee BS (2011) Event detection in Twitter. *Proc. 5th Internat. AAAI Conf. Weblogs Social Media* (AAAI Press, Palo Alto, CA), 401–408.

Xia C, Schwartz R, Xie K, Krebs A, Langdon A, Ting J, Naaman M (2014) CityBeat: Real-time social media visualization of hyperlocal city data. *Proc. Internat. Conf. World Wide Web* (ACM, New York), 167–170.

Xu Z, Ru L, Xiang L, Yang Q (2011) Discovering user interest on twitter with a modified author-topic model. Proc. IEEE/WIC/ACM Internat. Conf. Web Intelligence Intelligent Agent Tech., vol. 1 (IEEE, Piscataway, NJ), 422–429.

Ye M, Liu X, Lee WC (2012) Exploring social influence for recommendation: A generative model approach. *Proc. 35th Internat. ACM SIGIR Conf. Res. Development Inform. Retrieval* (ACM, New York), 671–680.

Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surveys* 52(1):1–38.

Zhang C, Zhou G, Yuan Q, Zhuang H, Zheng Y, Kaplan L, Wang S, Han J (2016) Geoburst: Real-time local event detection in geo-tagged tweet streams. *Proc. Internat. ACM SIGIR Conf. Res. Development Inform. Retrieval* (ACM, New York), 513–522.