Event Duration Detection on Microblogging

Yi-Shiang Tzeng*, Jyun-Yu Jiang† and Pu-Jen Cheng‡
Department of Computer Science and Information Engineering
National Taiwan University, Taiwan
{*r00944020, †b98902114}@ntu.edu.tw, ‡pjcheng@csie.ntu.edu.tw

***Abstract*—Microblog users often post what they observe in the surroundings, making it possible to use such microblog data to perform event detection. In this paper, we propose an issue of event duration detection and choose rain as our target event. Our goal is to construct an online virtual weather station, which reports local weather condition such as rain with the microblog data. Different from previous work focusing on earthquake, rain is a relatively minor event and may continue for a period of time. The virtual station, therefore, needs to detect not only when it starts but also when it finishes. The system trains a classifier to extract the event of interest. A wavelet-based method and aging-based method are proposed to detect the beginning of the rain event and its duration, respectively. Our experiments are conducted on real data, collected from Twitter and online weather stations. The results of experiments show the feasibility of virtual weather system. Our user behavior analysis also explains why the system works.**

***Keywords*-Event Duration; Event detection; Microblog;**

I. Introduction

With the popularity of microblogging service (e.g., Twitter), more and more people share what happens in their surroundings on the internet, including weather information such as the posts of "something is shaking my building like an earthquake" and "it's raining outside". Twitter[1], as an increasingly popular microblog service provider, already has 127 million active users up to 2012. 54% of users access Twitter through mobile devices and therefore grouping as a social sensor network to report what happen around in real time (e.g., rain, typhoon).

Owing to the real-time feature of Twitter, event detections by using Twitter messages (tweets) have caught much attention from recent research studies. Their researches often focus on the detection of busty feature of the event. However, real-world events (e.g., rain, snowstorm, and typhoon) might last for a period of time (e.g., minutes of an hour, hours of a day). Or it might happen one by one with a period of time break between them, leading it to happen sequentially for a period of time (e.g., minutes of an hour, hours of a day, days of a month). In this paper, we mainly focus on the duration of detected rain event.

Before travelling to a place, people often would like to know the weather there (e.g., rain). Unfortunately, most weather stations only provide weather forecast for cities or famous scenic spots. There are no weather data for local areas which lie in the city. However, the forecast for whole city is summarized from different regions. It cannot reflect regional variation. For example, in a city, it may rain in some places while it may not rain in other places; or some places may be located at the boundary between two cities. It is, therefore, difficult for people to obtain exact weather conditions for arbitrary places. Fortunately, data from increasingly popular online social network site (e.g., Twitter) making it possible to real-time report the rain event which happened anywhere at any time.

The goal of this paper is to detect the duration of rain event by using microblogging data (e.g., tweets). In the literature, several approaches are proposed to detect events from documents (e.g., the Topic Detection and Tracking task [1]) or blogs (e.g., Facebook[2]). However, they might not be applicable to deal with the detection of the rain events from microblogs. Microblogs are often shorter than documents and blogs. Twitter limits to 140 characters for each post while Facebook accepts up to 60,000 words and average length of documents in the TREC collection[3]. Also, microblog is frequently updated. Therefore, conventional statistic-based term weighting strategies like frequency might not reliable in analyzing microblog contents.

A very recent study, a real-time event detection system is proposed to detect earthquake by using Twitter data. Similar to their work, Twitter data are also used by us to detect the duration of rain event. However, features of rain are quite different from those of earthquake. Thus their proposed method might not be directly applied to our work. The differences between the rain and the earthquake are as follows: 1) the duration of rain event is often much longer. It is therefore worth studying the duration of rain event. 2) the influenced regions by rain event are much irregular while earthquake acts like a ripple, the intensity is spread out and decreasing from the center of earthquake. 3) the rain event is relatively common and might draw less attention. There are probably not many posts talking about the rain event. Thus it could be more difficult to detect the rain event via tweets than to detect the earthquake event.

This paper presents an online virtual weather station for detecting the duration of rain event. Given a time and a

[1]*Twitter*, a social networking service and microblogging service. https://twitter.com/

[2]*Facebook*, a social networking service owning over 900 million active users. https://www.facebook.com/

[3]*The Text REtrieval Conference (TREC)* is also more than thousands, yearly workshop providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. http://trec.nist.gov/

location, the system first collects potentially rain-related tweets that are posted by users near the location. Then it trains a classifier to filter out those tweets which are not related to the rain event. The number of the rain-related tweets in each time slot is counted and these numbers from all time slots then represent as a time series data. Aging theory [2] is then combined with wavelet and threshold methods to detect, respectively, the burst and the end of the rain event to determine the duration of rain event. Noted that our method can overcome the problem of data sparseness and avoid high false alarm or low recall. Finally, we conduct the experiments on real data, which are collected from Twitter and 13 U.S. online weather stations. The experimental results show the feasibility of the proposed system. Besides, we also investigate how soon users post tweets when they notice the rain events, and study whether the way user response to the rain event is the same to other events.

In the rest of this paper, we first make a brief review on event detection in Section II, and specify the problem in Section III. Virtual weather system is proposed in Section IV. The experiments and discussions are presented in Section V. Finally, in Section VI, we present our conclusions.

II. Related Work

With the growth of microblogging, event detection becomes more and more popular. Many algorithms have been performed on various datasets, such as documents, web pages and microblogs. Conventional event detection and tracking related researches mainly focus on documents(e.g., Topic Detection and Tracking [1] and Online Event Detection [3] [4]). By single-pass clustering(or incremental clustering), they calculate the similarity between the new coming document and existed ones to determine whether a new event/new branch is generated or not. Unfortunately, such clustering method can not be transplanted to the microblogging data directly. Unlike normal documents(e.g. web pages and blog articles), which contain hundreds to thousands or even more of words, the length of a post in microblogs is often much shorter,(e.g. One tweet contains 140 words at most on *Twitter*). It is, therefore, difficult to precisely estimate the similarity among these short documents, resulting in the failure of such method. Therefore, finding the boundaries of events is an important issue. Furthermore, we cannot expect that there are many event related documents or tweets all the time, especially when event goes on for a while. It makes the problem become harder.

In microblog, because of the short text characteristic, users usually use more precise words to convey their thoughts [5] . Therefore, we can detect events in the respect of words by monitoring their usages. [6] utilizes Twitter data to detect and locate earthquake in Japan by proposing a probability-based approach. [7] adopts aging theory to detect emerging terms, then groups these terms to a topic. Other than microblog, researchers also apply similar idea to detect events on short text. For instance, [8] analyzes Yahoo![4] query log for tracing the trojectory of a storm.

Except of directly doing event detection in time domain, some researchers also analyze the time series data in frequency domain. [9] collects 806,791 English news stories and analyzes all words both in time and frequency domains by Fourier transform. It then divides the detected event into important/minor and period/aperiod classes. In [10] and [11], the words of tweets on Twitter or tags of photos on Flickr are considered as the source of energy. The events are detected by using the wavelet transform to analyze the energy distribution. Although such methods can do well on signals changing suddenly (which often represents something happens); however, for the rain event, it only works partially, since we need not only detect the beginning of rain but also ending point, which usually has no obvious variation.

Although there are many existing researches making effort on event detection, all of above do not focus on event duration detection. Moreover, because rain is a minor and usual event, such characteristic makes this task become harder.

III. Problem Specification

Each Twitter user's post is called a "tweet". We define a tweet T_i as a 5-tuple $T_i = \{C(T_i),\ U(T_i),\ lo(T_i),\ la(T_i),\ t(T_i)\}$, where $C(T_i)$ and $U(T_i)$ are the content and publisher of tweet T_i. $lo(T_i)$ and $la(T_i)$ represent the longitude and latitude of the location where tweet T_i is posted, respectively. $t(T_i)$ denotes the published time.

Given a time t and a location (specified by longitude x and latitude y), we'd like to build a virtual weather station W_j, which is defined as $W_j = \{lo(W_j), la(W_j), E(W_j, t)\}$, where $lo(W_j)$ and $la(W_j)$ stand for the longitude and latitude of station W_j, respectively. $lo(W_j)$=x and $la(W_j)$=y. $E(W_j, t)$ indicates the weather condition in location (x,y) at time t. $E(W_j, t) = 1$ if it is raining; otherwise, $E(W_j, t) = 0$.

More specifically, suppose function $dist(T_i, W_j)$ gives the distance (varied from 0 to 1) between tweet T_i and station W_j based on their longitudes and latitudes. For virtual weather station W_j, our problem is how to effectively estimate $E(W_j, t)$ by the tweets $\{\ T_i\ \}$, where $t(T_i) \leq t$ and $dist(T_i, W_j) < \delta$. δ is a given threshold.

IV. The Virtual Weather System

The system contains two primary stages:"Tweets Filtering" and "Event Detection", as shown in Figure.1. In the first stage, we try to find out the tweets $\{\ T_i\ \}$ actually related to the rain event from all potential tweets (i.e., $dist(T_i, W_j) < \delta$ and $t(T_i) \leq t$) according to $C(T_i)$ and $U(T_i)$. In the second stage, we transform the numbers of the collected rain-related tweets into energy domain and then try to model the life cycle of a rain event to estimate $E(W_j, t)$ based on the concept of aging theory [2].

[4] *Yahoo!*, http://www.yahoo.com/

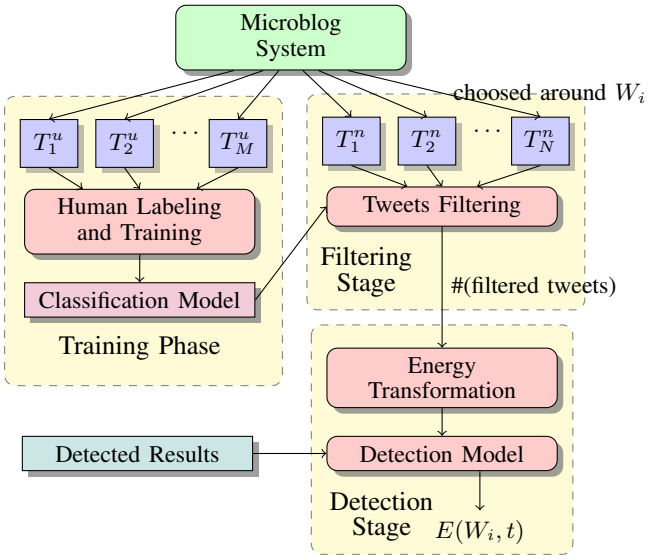


Figure 1. The flowchart of the system

IV-I. Tweets Filtering

In this stage, we want to exclude the tweets which are not "rain-related" to make our detection system more robust. The tweets fed to the system are simply classified as two classes, "rain-related" and "not rain-related".

We propose a method to find out the tweets discussing about the "rain" events. The method first applies heuristic rules to filter out those tweets not containing predefined rain-related keywords. But rain-related words might have multiple meanings in the natural language. For example, there is a song "Rain" recorded by the famous band "Beatles". Moreover, it is not necessary that the time a user talking about rain will be the time the rain happens. For example, people may post some tweets like "it's about to rain" or "the rain last night made me annoyed". Although such tweets are truly concerned about rain, for our application they still need to be considered as noises.

Considering the above factors, we find it essential to filter out the noises. Otherwise, we may make false alarms easily. Our filtering method, therefore, adopt Support Vector Machine (SVM) [12] as a classifier to classify tweets into "rain-related" and "non-rain-related" sets. We consider SVM because it is a discriminative model producing superior classification performance in many case.

For each tweet T_i, we extract text features from its text space $C(T_i)$ and author features from its author space $U(T_i)$. Text features can be divided into statistical features and semantic features. Statical features capture the statistical information of a tweet, including the length of the tweets, the frequencies and positions of predefined rain-related keywords "rain" and "raining", respectively semantic features are the top 4000 content words chosen by χ^2 statistic [13] from the training data. Author features keep the statistical information about a user, includeing the total number of tweets she posts, number of tweets she posts during the rain events and number of tweets she posts containing keywords "rain" and "raining". The author features are supposed helpful if there are some users worth trusting.

IV-II. Event Detection and Modelling

In this stage, we use the cleaned data obtained from previous stage and then calculate the number of tweets along time for representing a time series data. To estimate $E(W_j, t)$ for each weather station W_j at time t, we apply the concept of aging theory introduced by [2] to model the life cycle of a rain event, making it possible to determine event boundaries.

A. Energy Transformation

Aging theory considers an object as an organism. By taking the suitable resources as nutrition, the object obtains the energy it needs to develop. With the variation of the energy, the object goes through the four statuses of birth, growth, decay and finally die. In this paper, we consider rain events as our organisms and take rain-related tweets as clues for detect the event. When it starts to rain, people may be interested in the event. As time goes by, they are gradually used to the event and don't pay attention to it. Such phenomenon implies that the energy of rain event changes by time and we therefore use aging theory to model the process.

1) Nutrition: One possible way to calculate nutrition for a given time stamp is treat every tweet as identical and use the number of tweet as nutrition, but there is still much information worth to be considered. Actually, not every tweet related to rain is helpful. For instance, one user talking about rain in Los angles has little chance to be concerned with rain in New York, since twitter user tends to post what happens beside them. Therefore, it is essential to give different weight for the relevant tweets.

Different tweets T_i contribute nutrition at different levels. So they should be assigned different weights according to their attributes. For each tweet T_i, we define the weight of T_i as $P(E(W_j, t) = 1 \mid T_i)$, which can be extended by the Bayesian theorem as follow:

$$P(E(W_j,t)=1 \mid T_i) = \frac{P(T_i \mid E(W_j,t)=1)P(E(W_j,t)=1)}{P(T_i)},$$

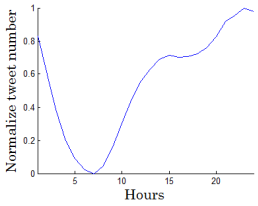
where three items need to be approximated. $P(T_i \mid E(W_j, t) = 1)$ models relationship between rain-related tweet T_i and rain event $E(W_j, t)$. $P(E(W_j, t) = 1)$ denotes the prior probability of the rain event happening in station W_j at time t, and $P(T_i)$ implies the prior probability of tweet T_i being posted.

$P(T_i|E(W_j, t) = 1)$: The item represents the probability of tweet T_i being posted given that it is raining in station W_j at time t. To estimate it, we assume that such probability is related to the distance between tweet T_i and station W_j. As [14] finds that frequency distribution of users varies as a power of their distances to

an event center, such distribution follows a power law. We, therefore, define a function $dist(T_i, W_j)^\alpha$, $\alpha \leq 0$ for estimating the probability. The function is proportional to the probability density function of power-law distribution. When α is equal to zero, tweets T_i are viewed as identical. Lower α emphasizes the importance of distance and can narrow down the scope of nearby regions.

$P(E(W_j, t) = 1)$: On the basis of meteorological knowledge, the weather condition of a region varies by season. The probability can be estimated by training data. However, as in the experiments our 3-month training data is too small to approximate the factor, we assume it is uniformly distributed and can be ignored.

$P(T_i)$: $P(T_i)$ is related to temporal factor. From the Figure 2(a), we find that the amount of the tweets posted in daytime is greater than that at midnight. To address these issues, we make use of the number of tweet T_i posted at around time $t(T_i)$. Note the number of rain-related twitters T_i posted over time is quite different from the number of whole tweets posted over time (We will explain the details in Section V.C). In our system, for each tweet T_i, $P(T_i)$ is attached a temporal weight $tmp(T_i)$, which is the number of twitters T_i posted at around time $t(T_i)$. We learn $tmp(T_i)$ from training data.



(a) Tweet distribution in a day

(b) Tweet distribution in third station

Figure 2. Tweet distribution of Twitter

By combining all the factors mentioned above; we define the weight of tweet T_i for station W_j as follow:

$$weight(T_i) = \frac{(dist(T_i, W_j)) + 1)^\alpha}{tmp(T_i)}$$

and nutrition in time t_k is calculated as

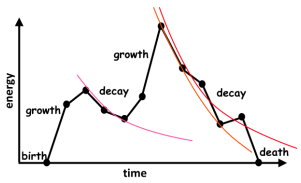
$$Nutrition(t_k) = \sum weight(T_i), \text{ for each } T_i, t(T_i) = t_k$$

2) Energy: We define that energy as the expected value of nutrition. A tweet is effective not merely at the time it is posted, its effect should last for a while and decrease by time. Nutrition is similar. We assume that probability of nutrition becoming out of date in every time slot is uniformly distributed. Consequently the probability of nutrition that is still effective can be modelled by a exponential function. Based on this assumption, we derive our energy function as following and set β as 0.85

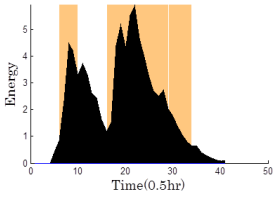
$$Energy(t_k) = \beta \times Energy(t_k - 1) + Nutrition(t_k)$$

Aging theory based Model

Now we have the energy distribution over the whole time line. The time slot size is set to be 1 hour.



(a) Four statuses of Aging theory



(b) Energy distribution from Nov 28 2011 to Nov 29 2011 in station 1. Background with color indicates raining.

Figure 3. The four statuses of energy in Aging theory

From Figure 3(b), we find that under normal circumstance, energy is relatively low and smooth. The occurrence of an event will break the regularity. As a result, by capturing the rigid change, we are able to catch the beginning of the event. We define such change point as the birth status of the event.

To detect boundaries, we apply wavelet transform on the time series energy data. Wavelet transform is one popular way for variation detection. It transforms the signal from time domain into frequency domain through a multi-resolution way. We choose the Daubechies wavelet [15] in our model because it can be applied on length-limited and discrete signal easily. We adopt the idea of H-measure proposed by [10] to calculate the energy entropy. As the following formula shows, for a signal s, H-measure calculates the energy in different resolutions ($E_{resolution_j}$), calculates their entropy (the numerator), and finally normalizes by maximum possible entropy ($Entropy_{max}$).

$$H(s) = \frac{-\sum \frac{E_{resolution_j}}{E_{total}} \log \frac{E_{resolution_j}}{E_{total}}}{Entropy_{max}}$$

Then as Figure 4 shows, we calculate the H-measure for each of the two time windows, and compute $\max(\frac{H_{small} - H_{large}}{H_{large}}, 0)$. If there is something happening, H-measure of small window turns into large while the large one remains relative small. We get a larger value. However, the signal bursts not only when energy is in increasing phrase but also sometimes in decreasing phrase. To solve the problem, we compare the original energy between the two time windows to filter out the burstiness in decreasing phrase.

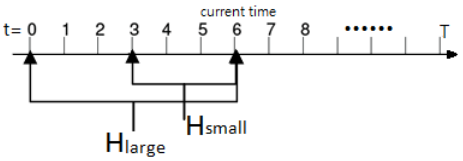


Figure 4. Illustrate for beginning detection

When one event occurs, it's nearly impossible for the energy to reach a maximum immediately. People requires some time to make responses. We consider such period, which keeps getting energy, as the growth status. In our model, events in this status will never be declared death.

After reaching the maximum of energy, the event enters the decay status and energy starts to decrease. For a minor event, the energy curve is often not smooth due to the data sparseness problem. Therefore we assume energy declines in an exponential way in this status. By the technique of exponential regression, we can solve an optimal function that fits the data best in the decay status. The optimal function is updated when new data comes (Figure 3(a)). At first time, we have only one point so there is nothing to learn, but in the next time slot, we have two points and the first version of the optimal function can be solved out. In the following n-steps in the decay status, we do the similar thing. The expected energy is returned from the optimal function. Differing from using the original energy, using the expected energy has additional benefits: Time factor is considered implicitly. Because the expected energy can be used to predict the missing signal due to the data sparseness problem. Moreover, because the exponential function decreases very fast, it can also fit dramatically-decreasing signal well.

The situation discussed above is an ideal case; however, in practice the energy signal seldom monotonically decreases. Most of the time energy signal goes up and down. Sometimes it even bursts again which might be caused by a new event. To distinguish the old event and the new event, we apply wavelet model here again. Once a significant change is found, we say that the current event is death and a new event is created; otherwise, the decay status will continue.

Once rain stops, the energy signal should return to a low and stable level, leading to the ending of the event. As Figure 3(b) shows, the change of energy in this status isn't as dramatic as that in the beginning, resulting in wavelets transform is not a good option for boundary identification here. Hence we propose a threshold-based method to determine the death of an event. To decide the threshold automatically, we assume that most of the energy is converged on the duration of the event, and energy that appears outside of the event is lower than energy inside the event. By predefining a pair of lower bound and upper bound (in this work we assume 70% energy is in the event so we set $(lb, up) = (0.65, 0.75)$), the threshold can be determined accordingly. The details of how we determine threshold are shown in Algorithm 2, and the overview of aging theory based model is displayed in Algorithm 1.

Algorithm 1 detectionModel($energy$)

$status \leftarrow$ NULL
$peak \leftarrow$ NULL
for $t = 0 \rightarrow T$ **do**
  if $isChangePoint(energy[t])$ **then**
    $status \leftarrow$ *BIRTH*
    $status \leftarrow$ *GROWTH*
  else if $status =$ *GROWTH*$\cap isMax(energy[t])$ **then**
    $status \leftarrow$ *DECAY*
    $peak \leftarrow t$
  else if $status =$ *DECAY* **then**
    $lambda \leftarrow ExpRegression(energy[peak : t])$
    if $newEnergy(lambda, t, peak) < threshold$ **then**
      $status \leftarrow$ *DEATH*
    end if
  end if
end for

Algorithm 2 getThreshold($energy, lb, ub$)

$energy \leftarrow sortHightoLow(energy)$
$totalEnergy \leftarrow sum(energy)$
$cumulateEnergy \leftarrow 0$
for $i = 0 \rightarrow length(energy)$ **do**
  $cumulateEnergy \leftarrow cumulateEnergy + energy[i]$
  if $\frac{cumulateEnergy}{totalEnergy} \leq lb$ **then**
    $low \leftarrow i$
  else if $\frac{cumulateEnergy}{totalEnergy} \leq ub$ **then**
    $up \leftarrow i$
  end if
end for
$threshold = mean(energy[low : up])$

V. Experiments

In this section, we first evaluate the performance of our model, including filtering, and raining detection, then analyze the influences of temporal factor. Finally, we do a user behavior analysis.

We use Twitter Stream API[5] to receive all public tweets real-time, such that we can simulate a on-line microblogging system with geographic information.

A. Evaluation of Filtering

In the filtering stage, we use the package LIBSVM [16] to train a SVM model with radial basis function(RBF) kernel. To ensure the good performance, we scale the features of training data into the range from 0 to 1, and do same operation on the testing data. Then we compare it to the results without scaling.

For the training step in the filtering stage, we totally label 6,320 tweets from Aug. 2012 to Sep. 2012 that contain "rain" or "raining". 2,797 tweets are related to the rain event; the remainders are not related but containing the keywords. We combine different features, including statistical, semantic and author features to train each model separately and do 5-fold cross validation to evaluate with measures of *precision*, *recall* and *F1-Score*.

From Table.I, we can see that the scaling preprocessing makes a great improvement in SVM with RBF kernel. For the feature selection, semantic features are the most effective. Although author features provide the best performance in recall, it performs the worst in precision. Such features generate too many false positives and do

[5]https://dev.twitter.com/docs/streaming-api

Feature Set	without scaling			with scaling		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
A	0.7041	0.8520	0.7710	0.7559	0.8056	0.7799
B	0.5833	0.6483	0.6141	0.5837	0.6455	0.6130
C	0.4721	0.9442	0.6295	0.4767	0.9378	0.6321
A + B	0.7560	0.7841	0.7698	0.7910	0.8063	0.7986
A + C	0.7252	0.8120	0.7661	0.7531	0.8077	0.7794
B + C	0.6187	0.6594	0.6384	0.6269	0.6312	0.6290
A + B + C	0.7500	0.7806	0.7650	0.7940	0.8031	0.7985

Table I
PERFORMANCE OF FILTERING, A MEANS SEMANTIC FEATURES, B MEANS STATISTICAL FEATURES, C MEANS AUTHOR FEATURES

$$\textit{Precision} = \frac{\text{the number of alarmed time slots with weather raining}}{\text{the number of alarmed time slots}}$$

$$\textit{Recall} = \frac{\text{the number of alarmed time slots with weather raining}}{\text{the number of time slots with weather raining}}$$

$$\textit{F1-Score} = \frac{2 \times \textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}}$$

$$Accuracy = \frac{\text{the number of time slots with correct prediction}}{\text{the number of all time slots}}$$

Figure 5. Four measures in evaluation of event detection

	baseline	wavelet base	raw data	SVM	T+SVM
precision	0.4972	0.2010	0.4764	0.5610	0.5528
recall	0.6253	0.6708	0.7196	0.7102	0.7254
F1-Score	0.5539	0.3093	0.5732	0.6268	0.6274
Accuracy	0.9299	0.7997	0.9226	0.9432	0.9424

Table II
PERFORMANCE OF DETECTION MODEL. T MEANS TEMPORAL WEIGHT. TOO LITTLE SENSOR STATION(4,5,9) ARE NOT USED TO COMPARE IN THIS TABLE

not benefit the performance when combined with others. Finally, we use statistical and semantic features to build the final classification model.

B. Evaluation of Event Detection

To evaluate the performance of the proposed event detection method, a baseline method is built up. The baseline method simply judges $E(W_j, t)$ as 1 if the number of the rain-related tweets posted at the time slot t is greater than a threshold, determined by Algorithm 2. Noted that such method is equal to do event detection on each time slot. We further implement a wavelet-based method [10] for comparison. This is because wavelet filtering is quite sensitive to signal changes, which often happen at the boundaries of events such as beginning and ending. The time slots between a boundary are labelled "raining".

We select thirteen weather stations in U.S. and treat the weather condition they post as ground truth. For each station, we crawl all the tweets around it from Nov. 01, 2011 to Jan. 31, 2012 (3 months). Table.IV summarizes their statistics. Evaluation is done by 3-fold cross validation with time slots set to be one hour. The performance is measured by four measures - *precision*, *recall*, *F1-score*, and *accuracy*. Precision here is defined as the number of the time slots in which our method and ground truth both agree "it is raining" divided by that only our method alarms. The other measures are also defined in Figure 5.

Table II shows the results in terms of the four metrics. Various settings of the proposed method are taken into account. "RAW" means the method counts all the tweets containing "rain" or "raining" in. The method denoted by "SVM" further performs tweet filtering, i.e., the first stage. "T" represents the temporal weight. Each method may combine different settings. From Table II, we find the wavelet-based method produces lower precision, even compared with the baseline. This is because such method is not suitable for detecting the ending of a rain event, where the signal is not always changed abruptly. Twitter filtering really reduces noisy input signals and is helpful in precision. But it could hurt recall to some extent. Consider F1-score, which puts balanced emphasis on precision and recall. The proposed method under different settings performs much better than the baseline and wavelet-based methods. Some improvements will be gained when certain factors or weights are considered. In our method with different setting, recall is much higher than precision because our method not only produces high energy during the rain events but increases the energy of the time slots that are adjacent to the rain events. Accuracy is quite high in the results. Actually, most of the time it doesn't rain near the thirteen weather stations.

We adjust the time slot size to see how the parameter affect final performance. The tendency in precision and recall are plotted in Figure 7. Smaller time slot size brings higher recall and lower precision. When enlarging time slot size, we probably collect more rain-related tweets in each time slot, leading to more reliable signals; however, it may make the signal changes not obvious at the boundaries so recall will decrease. We also examine how the performance varies when we change the β of energy function in Figure 6. When increasing the β, we get the higher performance since the problem of data sparse is improved. However, when β is larger than 0.8, the performance drop rapidly. It is because that too many false alarms are created.

Our system is not always feasible for every station, as shown in Table III. Take the third weather station as an example, where the population centralizes in certain subregions (See Figure 2(b)). Unbalanced energy distribution cannot actually reflect the weather condition in the centroid

station id	1	2	3	4	5	6	7	8	9	10	11	12	13
F1-score	0.6602	0.7351	0.2100	0.0513	0.3259	0.6703	0.5062	0.5664	0.0746	0.7368	0.7493	0.6399	0.5857

Table III
PERFORMANCE OF EACH STATION

station ID	#(Tweets)	Tweets containing "rain" or "raining"	Alarmed Tweets	tweet/day×1000	population distribution entropy	raining hours	daytime raining hours	night raining hours
0	3,246,499	3,298	1,720	11.5845	0.7794	62.00	44.25	17.75
1	4,275,156	3,983	2,083	11.0174	0.7369	104.50	84.50	20.00
2	4,385,449	4,228	2,243	11.9762	0.6739	52.00	40.25	11.75
3	1,113,344	883	550	2.5402	0.8909	35.50	28.00	7.50
4	1,083,468	849	518	2.4717	0.9122	31.50	23.25	8.25
5	2,810,782	4,034	2,675	7.4618	0.9061	66.25	46.50	19.75
6	1,201,155	2,023	1,164	3.1549	0.5387	79.50	44.75	34.75
7	2,202,675	2,624	1,005	5.4941	0.7983	137.50	82.25	55.25
8	512,707	1,311	503	1.5720	0.7774	23.50	12.50	11.00
9	1,168,641	1,297	607	3.0482	0.8014	120.75	82.25	38.50
10	2,504,224	2,685	1,824	6.2464	0.8719	189.25	130.00	59.25
11	760,016	1,572	649	2.5543	0.8818	126.75	79.00	47.75
12	1,817,031	3,919	2,013	4.5148	0.9136	89.00	50.00	39.00

Table IV
STATISTICAL DATA OF WEATHER STATIONS

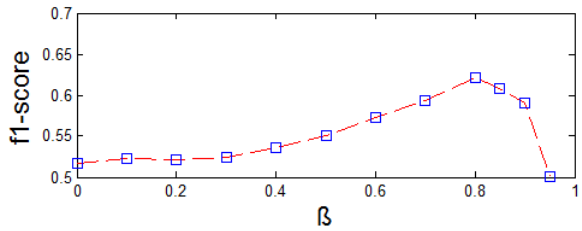


Figure 6. Influence of β

of a region, leading F1-score in this station to 0.25. On the other hand, our performance is affected by the number of Twitter users. Since we view them as social sensors, sufficient number of the users is essential for our system. The top 3 stations with most rain-related tweets can reach 0.7351, 0.7493, and 0.6602 in F1-score, respectively, while the last 3 only 0.3259, 0.0746, and 0.0513.

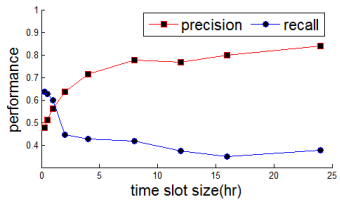


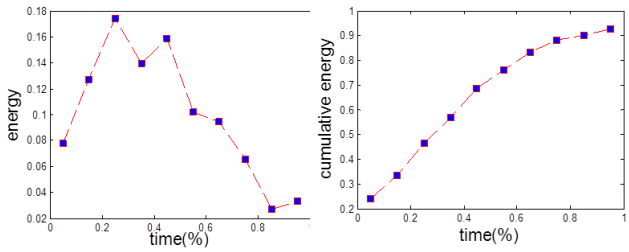
Figure 7. Influence of size of time slot

C. User Behavior Analysis on Rain Events

Here we want to (1) realize when the Twitter users post weather information on Twitter once the rain events get their attention and (2) study how the temporal factor affect the user behavior, i.e., sharing weather information with others.

First, we collect 22 rain events that last at least five hours from all the weather stations and calculate their energy distribution over time. Figure 8(a) shows the results, by which we find 18.1% of energy appears before the rain events; that is, some rain-related tweets are posted earlier than the events. 74.6% of energy converges on the duration of rain; that is, most rain-related tweets are posted during the events. 7.3% appears after the events. Figure 8(a) shows such distribution. Energy reaches a maximum at the first 30% of the time. Figure 8(b) shows cumulative energy distribution. 80% of energy is cumulated before the 60% of the time in average. The energy distribution observed is roughly similar to what we imagine and meets the assumption of the aging theory. As rain is affected by many factors such as terrain and wind direction, if it doesn't rain in one region but rains in its nearby regions, we probably receive event signals before the rain event actually happens in the region. That's why 18.1% of energy appears earlier.

Second, we compare the normalized numbers of total tweets with those of rain-related tweets posted over 24 hours, as shown in Figure 9, where 3-month tweets are collected from 13 weather stations and the average numbers are both smoothed and normalized between 0 and 1. We find that the two curves look similar except time difference. People post weather information earlier than other information in the morning. This might be because they are more concerned about weather condition in the daytime. Interestedly, the total number of tweets grows in the evening and reaches the maximum at midnight; however, people pay less and less attention to weather condition from 6:00 pm. Based on this observation, we find that whether a user posts weather condition is related to the time she makes a post and is not proportional to the total number of the tweets.



(a) energy distribution in rain event (b) cumulative energy distribution

Figure 8. The energy distribution

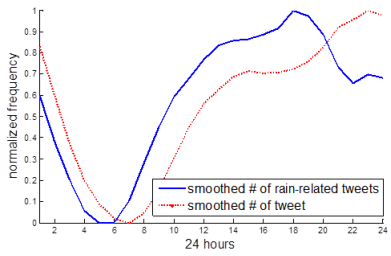


Figure 9. User behavior analysis:Total tweets vs. Rain-related tweets

VI. Conclusions

In this paper, we propose an issue of event duration detection on microblogging. Unlike the event detection, which focus on whether an event is detected or not, event duration detection cares when the event begins and when the event finishes simultaneously. To judge the performance of our model, we select "rain" as our target event for experiment.

The framework contains two stages. In the "Tweets Filtering" stage, we propose a method using both text and author information as features for extracting relevant tweets to the rain events. In our experiment, the method can find out around 80% positive tweets with low false alarm. In the "Event Detection" stage, we propose a event detection model based on aging theory. We first calculate the energy for each time slot by considering temporal factor and apply wavelet technique to detect the beginning of an event. Then we use the exponential function to model the decay status and finally apply a threshold-based method to determine the ending of an event.

In our experiments, we show that our method is feasible for detecting the duration of rain. Nevertheless, our performance is limited and influenced by several factors such as time, population and population distribution. As a result, we need to give different weights for different tweets according to their attributes. Sparse data or unbalanced population may cause the bad performance. We also show that directly doing event detection in each time slot can't use to instead event duration.

In the final we analyze the user behaviour for rain events. In our study, most users post rain-related tweets in the duration of rain and some even talk earlier than that event happens. We also find that users get different attention at different time.

Acknowledgment

This work was supported by National Science Council under 101-2815-C-002-097-E.

References

[1] J. Allan, *Topic detection and tracking: event-based information organization*, 2002, vol. 12.

[2] C. C. Chen, Y.-T. Chen, and M. C. Chen, "An aging theory for event life-cycle modeling," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2007.

[3] Y. Yang, T. Pierce, and J. Carbonell, "A study on retrospective and on-line event detection," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.

[4] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.

[5] R. Long, H. Wang, Y. Chen, O. Jin, and Y. Yu, "Towards effective event detection, tracking and summarization on microblog data," *Web-Age Information Management*, 2011.

[6] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*, 2010.

[7] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, 2010.

[8] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak, "Spatial variation in search engine queries," in *Proceedings of the 17th international conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 357–366. [Online]. Available: http://doi.acm.org/10.1145/1367497.1367546

[9] Q. He, K. Chang, and E.-P. Lim, "Analyzing feature trajectories for event detection," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '07, 2007.

[10] W. Jianshu and L. Bu-Sung, "Event detection in twitter," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[11] L. Chen and A. Roy, "Event detection from flickr data through wavelet-based spatial analysis," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009.

[12] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Machine Learning: ECML-98*, pp. 137–142, 1998.

[13] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization." Morgan Kaufmann Publishers Inc., 1997.

[14] E. Yom-Tov and F. Diaz, "Out of sight, not out of mind: on the effect of social and physical detachment on information need," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.

[15] I. Daubechies, *Ten lectures on wavelets*, 1st ed. Society for Industrial and Applied Mathematics, Jun. 1992.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2011.